A KNOWLEDGE BASE SYSTEM APPROACH TO INSPECTION SCHEDULING FOR FIXED OFFSHORE PLATFORMS

Sarah M C Peers

Submitted for the Degree of
Doctor of Philosophy in Mechanical Engineering
University of London

University College London
1998

ABSTRACT

In the offshore oil and gas industry in the UK, one of the most common forms of structure is the fixed steel jacket type of offshore platform. These are highly redundant structures subject to many random or uncertain factors. In particular, they are subject to uncertainties in the load distribution through the components, and to time-varying and cyclic loads leading to deterioration through fatigue. Operators are required to ensure the integrity of these structures by carrying out periodic inspections and repairing when necessary.

Decisions on inspection, repair and maintenance (IRM) actions on structures involves making use of various tools and can be a complex problem. Traditionally, engineering judgement is employed to schedule inspections and deterministic analyses are used to confirm decisions. The use of structural reliability methods may lead to more rational scheduling of IRM actions. Applying structural reliability analysis to the production of rational inspection strategies, however, requires understanding the inspection procedure and making use of the appropriate information on inspection techniques. There are difficulties in collecting input data and the interpreted results need to be combined to form a rational global solution for the structure which takes into account practical constraints.

The development of a knowledge base system (KBS) for reliability based inspection scheduling (RISC) provides a way of making use of complex quantitative objective analyses for scheduling. This thesis describes the development of a demonstrator RISC KBS. The general problems of knowledge representation and scheduling are discussed and schemes from Artificial Intelligence are proposed. Additionally, a system for automated inspection is described and its role in IRM of platforms is considered.

A RISC System integrating suitable databases with fatigue fracture mechanics based reliability analysis within a KBS framework will enable operators to develop rational IRM scheduling strategies.

ACKNOWLEDGEMENTS

I would like to acknowledge past and present colleagues. In particular, I am grateful to Dr Dharmavasan, Dr Jacob Kam, Dr Michael Faber, Professor Bill Dover, Professor Dave Broome, Dr Henry Ellingworth, Ming Xi Tang, Dr Onno Dijkstra of TNO Building and Construction Research and Roberto Cazzulo of the Registro Italiano Navale for providing me with many points to ponder over. I would also like to thank Professor Roy Collins for his patient reading of the draft versions of this thesis.

And many thanks to Jeremy and my parents:

> *¡Hay que ver la reputación de vago que tengo aquí! [...] Allá en España, «hay años en los que no está uno para nada». Llega de pronto un momento decisivo, y entonces el español trabaja como una fiera durante quince días. Es posible al cabo de esos quince días el español haya hecho tanto como lo que hace un inglés en un año entero.*
>
> *[...] yo muchas veces pienso que el progreso no debe realizarse convirtiendo a los hombres en máquinas, sino haciendo máquinas tan perfectas que parezcan organismos humanos.*
>
> ***Julio Camba***

> *La vida es duda,*
> *y la fe sin la duda es sólo muerte.*
>
> ***Miguel de Unamuno***

TABLE OF CONTENTS

1 INTRODUCTION 17

1.1 **Background** 17

1.2 **History of Offshore Structures** 18

1.3 **Motivation** 19

1.4 **Glossary** 22

1.5 **Nomenclature** 25

1.6 **Organisation of the Thesis** 28

1.7 **Summary of Work** 30

- Reliability-based Inspection SCheduling for fixed offshore platforms (RISC) 30
- Automated Image Reconstruction using Expert Systems (AIRES) 31

1.8 **Summary** 31

2 INSPECTION, REPAIR AND MAINTENANCE OF OFFSHORE STRUCTURES 32

2.1 **An Overview of Current Procedures** 33

2.1.1 Guidelines and Legislation 33

2.1.2 Participants 35

2.1.3 Selection Criteria 35

2.1.4 Analysis Methods 38

2.1.5 Scheduling Constraints 39

2.2 **Inspection of Tubular Joints** 40

2.2.1 Inspection Techniques 41

2.2.1.1 Visual Inspection 41

2.2.1.2 Magnetic Particle Inspection 41

2.2.1.3 Eddy Current 42

2.2.1.4 Ultrasonic Characterisation 42

2.2.1.5 Alternating Current Potential Drop and AC Field Measurement 42

2.2.1.6 Flooded Member Detection 42

2.2.2 Inspection Data 43

2.2.3 Inspection Procedures 45

2.2.4 Other Tasks 45

2.2.5 Summary 46

2.2.6 Operators' Future Requirements 46

2.3 **Fatigue Fracture Mechanics of Fixed Offshore Structures** 47

2.3.1 The Process of Fatigue 48

2.3.1.1 Loading Modes 49

2.3.2 Modelling Fatigue 50

2.3.2.1 Miner's Rule 50
2.3.2.2 Stress-Life Curves 50
2.3.2.3 Loads and Stress Intensity Factors 51
2.3.2.4 The Erdogan-Paris Relation 53
2.3.3 Fatigue Life of Offshore Structures 54
2.3.3.1 Hot-Spot Stress and Stress Concentration Factors 54
2.3.3.2 The Effect of Corrosion 55
2.3.4 Summary 56
2.4 Current IRM Scheduling and Planning Methodologies 57
2.4.1 Failure Mechanisms 57
2.4.2 Inspection Reliability 57
2.4.3 Load and Stress Histories 58
2.4.4 Other Factors 59
2.4.4.1 Repair Actions 59
2.4.4.2 Costs and Measures of Utility 60
2.4.4.3 Gross Errors and Human Factors 60
2.4.5 Current Work in Inspection Planning 60
2.4.5.1 Worst Case Crack Growth 61
2.4.5.2 The Relative Operating Characteristic (ROC) Method 62
2.4.5.3 Fatigue Reliability 63
2.4.5.4 The MSG -3 Method 63
2.4.5.5 Risk and the ALARP Principle 64
2.4.5.6 Structural Reliability Methods 65
2.5 Discussion and Conclusions 66

3 RELIABILITY ANALYSIS OF OFFSHORE STRUCTURES 68
3.1 Why Reliability for Offshore Structures? 68
3.1.1 Types of Uncertainties 70
3.2 General Background to Reliability Analysis 70
3.2.1 Uncertainties in Structures 71
3.2.2 The Limit State Function 72
3.2.2.1 The Fundamental Case 73
3.2.2.2 A Definition of the Reliability Index 74
3.2.3 Levels of Structural Reliability Analysis 75
- Level I 75
- Level II 75
- Level III 75
- Level IV 76

3.2.3.1 Overview of Level II Analysis 76
3.2.3.2 The General Level II Method 78
3.2.3.3 First and Second Order Reliability Method 79
3.2.4 Extending Structural Reliability Analysis 80
3.2.4.1 Updating the Reliability Index 80
3.2.4.2 Systems Reliability 81
■ β-Unzipping 84
■ Branch-and-Bound 85
■ A Pragmatic Approach 86
3.2.5 Applications of Reliability Analysis in the Offshore Industry 87
3.2.5.1 Design Codes 88
3.2.5.2 Evaluation of Designs 88
3.2.5.3 Probabilistic Analysis For Planning 88
3.3 Structural Reliability Analysis for IRM Planning 89
3.3.1 A Limit State Function for IRM Planning 90
3.3.2 Optimal Maintenance Planning 92
3.3.2.1 Overview of the Use of Decision Theory 92
3.3.2.2 Identification of Optimal IRM Plans 94
3.3.3 Modelling of Inspection and Repair 94
3.3.3.1 Inspection Events 95
3.3.3.2 Modelling of Repairs 96
3.3.3.3 Calculation of Costs and Sensitivities 97
3.3.4 Evaluation of Updated Reliability 98
3.3.4.1 Use of Inspection Results 99
3.3.5 Structural Analysis in RISC 99
3.3.5.1 Loading Analysis with ULDAN 100
3.3.5.2 Fatigue Fracture Mechanics Module (CFA) 101
3.3.5.3 RISCREL 102
3.4 Summary 103

4 KNOWLEDGE BASE SYSTEMS FOR IRM PLANNING 105
4.1 Review of Artificial Intelligence Concepts 105
4.1.1 Pattern Recognition Techniques 107
4.1.1.1 Issues of Pattern Recognition 107
4.1.1.2 Neural Networks 108
4.1.2 Expert Systems and Knowledge Base Systems 110
4.1.2.1 Knowledge Bases 110
4.1.2.2 User-Interface 111
4.1.2.3 Inference Engine 112

4.2 Issues in Knowledge base systems 113

4.2.1 Knowledge Representation Schemes 113

4.2.1.1 Layers of Knowledge 113

4.2.1.2 Logic 115

4.2.1.3 Production Rules 115

4.2.1.4 Semantic Nets 117

4.2.1.5 Schemas: Scripts, Frames and Objects 118

- Inheritance and Cross-referencing 120
- Stereotypes 120
- Summarising Schema 120

4.2.2 Reasoning and Induction Methods 121

4.2.2.1 Backward and Forward Reasoning 121

4.2.2.2 Models and Model-based Reasoning 122

4.2.2.3 The Blackboard Model 123

4.2.3 Modelling Uncertainty 124

4.2.3.1 Bayesian Methods 125

4.2.3.2 Fuzzy Logic 126

4.2.4 Knowledge Modelling for RISC 129

4.3 AI for Scheduling and Planning 130

4.3.1 The Scheduling and Planning Problem 130

4.3.2 Tree-Searching Techniques 132

4.3.2.1 Depth and Breadth First Methods 133

4.3.2.2 Best-First Search and Other Methods 135

4.3.3 Constraints-Based Planning 136

4.3.4 Applying Tree-Searching and Constraints Satisfaction to RISC 138

4.4 Review of Applications of KBS 139

4.4.1 Early Examples of Expert Systems 140

4.4.2 General Engineering Applications 141

4.4.3 AI based Planning and Scheduling Systems 143

4.4.4 Use of AI in the Oil Industry 145

4.5 Conclusions 146

5 IMPLEMENTATION OF THE RISC KBS 148

5.1 Requirements of an IRM planning system 148

5.1.1 Conceptual Model 149

5.1.1.1 Structural Knowledge Base 150

5.1.1.2 Analysis Procedures 151

5.1.1.3 Reliability-based Fatigue and Fracture Mechanics Analysis 151

5.1.1.4 Scheduling and Planning 152
5.1.2 Outline of the Functional Subsystems 152
5.1.3 User Interfaces 153
5.1.3.1 Interactive Use 154
5.1.3.2 Reports and Schedules 154
5.1.3.3 Modifications of Knowledge Bases 155
5.2 A Scheduling Model 155
5.2.1 Outline of Scheduling Procedure 155
5.2.2 A RISC System 156
5.2.3 The Ranking Algorithm 158
5.2.4 Analysis of Joints 159
5.2.4.1 Cost Evaluation of Maintenance Plans 161
5.2.5 Interpretation of Results 162
5.2.5.1 Heuristics-based Evaluation 162
5.2.5.2 Halting of Analysis 163
5.2.6 Combined Scheduling of Actions 163
5.2.6.1 Schedule Utility Function and Resource Deficits 164
5.2.6.2 Constraints Satisfaction Scheduling 165
5.2.6.3 Neighbouring and Similar Joints 166
5.2.6.4 User Modifications 167
5.3 Design and Data Structures for the RISC System 167
5.3.1 RISC Software Architecture 167
5.3.1.1 RISC System Tasks 168
5.3.1.2 RISC System Components 168
5.3.2 System Controller 171
5.3.2.1 Control of Rule Set Knowledge Sources 171
5.3.2.2 Interfaces to Analysis Modules 172
5.3.2.3 Database Interfaces 173
5.3.2.4 Documentation and Reports 173
5.3.3 View & Run Subsystem 173
5.3.3.1 Ranking KS 175
5.3.3.2 Maintenance Plan Formulation KS 175
5.3.3.3 Cost Evaluation KS 176
5.3.3.4 Propose Actions KS 176
5.3.3.5 Constraints Satisfaction Scheduling KS 176
5.3.4 Data Flow, Structures and Storage 176
5.3.4.1 Data Storage 177
5.3.4.2 Structural Objects 177

5.3.5 Set-Up, Observations and System Management 180
5.3.5.1 Set-Up Tasks 181
5.3.5.2 Observations Features 183
5.3.5.3 System Management Features 183
5.3.6 The User Interface and other Interfaces 184
5.3.6.1 Datafile Interface 184
5.3.6.2 The User Interface 185
5.4.1 Review of Software Tools 188
5.4.1.1 AI Programming Languages and Associated Tools 189
- Production Languages 190
- Functional Languages 190
- Hybrid Languages 190
- Object-Oriented Programming 191

5.4.1.2 Libraries and Miscellaneous Tools 191
- Class libraries 192
- Code Generators 192

5.4.1.3 Expert System Shells and Application Development Systems 192
- Production-Rule Development Systems 194
- Object-Oriented Development Systems 194

5.4.2 Requirements for Implementation 195
5.4.3 The KBS Development Tool, ProKappa 196
5.4.4 The RISC Demonstrator 198
5.5 Summary 199

6 A CASE STUDY 211
6.1 The Example Structure 211
6.1.1 Notes on the use of RISCREL 212
6.2 Modelling Cases for RISCREL 212
6.2.1 Geometry 213
6.2.2 Modelling of Repair 216
6.2.3 Material Properties 218
6.2.4 Loading 219
- WASR 220
- Weibull Method 220

6.2.5 Fatigue Crack Growth Modelling 221
6.2.5.1 SIF Method 221
- AVS 221
- TPM 222

6.2.5.2 2-Dimensional Crack Modelling 222
6.2.6 Modelling of Inspection 222
6.2.6.1 POD and POS 224
6.2.7 Costs Data 225
6.3 The RISC Analysis 226
6.3.1 The Tubular Components and Maintenance Plans 226
6.3.2 Execution of RISCREL 231
6.3.2.1 Updated Reliability 232
6.3.2.2 Cost Evaluation 232
6.4 Constraints Satisfaction Scheduling in the RISC System 235
6.4.1 Target Safety Levels 235
6.4.1.1 Effect on Maintenance Plans 238
6.4.2 Using the Scheduling Module 239
6.4.2.1 Resources for the Search Tree 239
6.4.2.2 Alternative Actions 240
6.4.3 Example Programmes of IRM Actions 241
6.4.3.1 Generating Schedules 243
6.4.3.2 Heuristics-based Searching and Multiple Constraints 245
6.4.3.3 Selection of a Programme 247
6.5 Discussion 248

7 AUTOMATED INSPECTION 259
7.1 Introduction to The AIRES System 259
7.1.1 The Electromagnetic ACFM Inspection Technique 260
7.1.1.1 The Electromagnetic Field Data 260
7.1.1.2 The ACFM Sensor 260
7.1.2 The Architecture and Requirements of the AIRES KBS System 262
7.1.2.1 Architecture of the Knowledge Base System 262
7.1.2.2 User Interface 263
7.1.2.3 The Vision Sensor 263
7.1.2.4 The Electromagnetic Sensor 264
7.1.2.5 Data Fusion 264
7.1.3 The Application and Phases of Work 264
7.2 The Blackboard System 265
7.2.1 The Development Tool, TRAM 266
7.2.2 Blackboard Concepts 267
7.2.3 Knowledge Sources 268
7.2.4 The Agenda 268

7.2.5 Interpretation of the Electromagnetic Sensor Data 269
7.2.5.1 Typical Theoretical electromagnetic Data 269
7.2.5.2 Typical Classification of Defects 270
7.2.6 The Signal Interpretation Procedure 270
7.2.6.1 Pre-processing 270
7.2.6.2 Flaw Detection 271
7.2.6.3 Flaw Classification 271
7.2.6.4 Feature Extraction 272
7.2.6.5 Grading/Output 272
7.2.7 Issues in Real ACFM Sensor Data Interpretation 272
7.2.7.1 Removal of noise 273
7.2.7.2 Peak/Trough Location Methods 274
- Simple Searching for Peaks and Troughs 274
- Location of Local Maxima and Minima 275
- Fitting Surfaces to the Magnetic Field Data 276

7.3 Knowledge Sources and Concepts for the Electromagnetic Sensor 276

7.3.1 Detailed Design of Concepts and Knowledge Sources 277
7.3.2 Electromagnetic Sensor KBS Concepts 278
7.3.3 Overview of Knowledge Sources 280
7.3.3.1 EM Sensor Control KSs 283
7.3.3.2 Electromagnetic Sensor Signal Interpretation KSs 284
7.3.3.3 Data Fusion KSs 285
7.3.3.4 User interface KSs 285
7.3.3.5 Future KSs 285
7.3.4 The Use of Neural Networks 287
7.3.5 Issues of Data Fusion 288
7.3.5.1 Asynchronous Use of Sensors 289
7.3.5.2 Combined and Irregular Flaws 289
7.3.5.3 Opportunistic Partial Interpretations 290
7.3.5.4 Data Fusion Concepts 291
7.3.5.5 Expert Knowledge as part of Data Fusion 291
7.3.6 Implementation and Testing 291
7.3.7 Tests for Phase I Detection and Categorisation 292
7.3.7.1 Theoretically Generated Data 293
7.3.7.2 Real Sensor Data Results 293

7.4 Concluding Remarks 294

8 CONCLUSIONS 296

8.1 Background to the Work 296

8.2 Summary of Achievements **297**

- Reliability based Inspection SCheduling for fixed offshore structures (RISC) 297
- Automated Image Recognition using Expert Systems (AIRES) 298

8.3 Future Work **299**

8.3.1 Analysis of the Structure 299

8.3.2 Scheduling Algorithms 301

8.3.3 An Integrated Monitoring, Inspection and Planning System 302

8.3.4 Other Applications 304

8.4 Final Remarks **305**

REFERENCES **306**

APPENDIX - PUBLISHED PAPERS **324**

LIST OF FIGURES

1.1	A typical jacket type platform	20
1.2	Typical node configurations for jacket type platforms	21
2.1	Current IRM procedure described by the Department of Energy	34
2.2	DEn IRM procedure making use of weighting factors	37
2.3	Typical fatigue crack growth	48
2.4	Mode direction and notation	49
2.5	Typical S-N curve	51
2.6	Use of SIF to calculate the actual stress at a point P	52
2.7	Typical fatigue crack growth versus local stress range	53
2.8	Multi-segment da/dN versus stress range ΔK	56
2.9	Crack growth and the a_{NDI} concept	61
2.10	Inspection planning using the a_{NDI} approach	62
2.11	A typical Receiver Operating Curve with Value Line	62
2.12	The ALARP principle	65
3.1	Loading and strength variations for two components	72
3.2	Normally distributed strength and loading random variables R and S	73
3.3	Linear failure surfaces (a) in the R, S space and (b) in the Z_1, Z_2 space	77
3.4	Tangent surfaces to the failure surface at point P_r	79
3.5	Components combined (a) in series and (b) in parallel	82
3.6	(a) Series - parallel and (b) parallel - series systems	83
3.7	Level 2 modelling of a structure	85
3.8	Fracture mechanics model for tubular joints	91
3.9	Semi-elliptical surface crack shape	91
3.10	Decision tree	92
3.11	Expected costs	98
3.12	Typical updated reliability index values	99
3.13	Structural analysis in RISC	100
3.14	The loading analysis procedure	101
3.15	The overall crack growth analysis procedure	102
3.16	The reliability analysis module, RISCREL	103
4.1	A simple neural network	108
4.2	Components of a KBS	112
4.3	Structure of a production rule	116
4.4	Example production rule	116

4.5	Example semantic net	117
4.6	A simple blackboard architecture	123
4.7	Membership values for the fuzzy set "Normal Loads for a brace"	127
4.8	Fuzzy sets (a) 'Heavy Loads' and (b) 'Light Loads'	127
4.9	A general search tree	132
4.10	A simple binary tree with 3 layers	133
4.11	Depth first search	134
4.12	Breadth-first search	134
4.13	An AND/OR tree	136
5.1	The RISC System	149
5.2	The scheduling procedure	156
5.3	A flowchart for a session	157
5.4	Calculation of ranking measure	159
5.5	Analysis of a single joint	160
5.6	"Inspection times" rule	161
5.7	Determination of inspection periods	161
5.8	Combined scheduling	164
5.9	RISC constraints satisfaction scheduling tree	166
5.10	RISC hierarchy of tasks	169
5.11	The use of intermediate files	172
5.12	Data flow in the RISC View & Run subsystem	174
5.13	Outline structural model	178
5.14	Structural objects for scheduling	179
5.15	Overview of set-up process	180
5.16	Datafiles processing	184
5.17	User interface (a) Views of structure, (b) Selection box, (c) Analysis control	186
5.18	User interface (a) Ranking, (b) Output of analysis results, (c) Scheduling	187
5.19	NoAction ProKappa rule	197
5.20	InspectNow ProKappa rule	197
5.21	The complete RISC System	198
5.22	Ranking module in the RISC Demonstrator	207
5.23	Display of RISCREL cost evaluation results	208
5.24	An initial schedule with the RISC Demonstrator	209
5.25	A usable and rational schedule after resource allocation	210
6.1	Case study tubular connections	216
6.2	Case study structure	227

6.3 Example joint identifiers 228
6.4 Reliability index values 233
6.5 Costs for joints with maintenance plans 234
6.6 Reliability index for joints (a) 13 and (b) 16 238
6.7 Search tree for the initial schedule 243
6.8 Heuristics-based search algorithm 247
7.1 The magnetic field **B** over a crack 261
7.2 Schematic diagram of the ACFM probe showing arrangement of coils 261
7.3 Knowledge base system structure for AIRES 262
7.4 Graphical user interface display 263
7.5 The structure of TRAM 266
7.6 $\mathbf{B}_y$ peaks and troughs for cracks and pits 270
7.7 Electromagnetic sensor KBS signal interpretation procedure 271
7.8 $\mathbf{B}_x$ data for two features 273
7.9 Mesh of points 274
7.10 Electromagnetic sensor KBS concepts 278
7.11 Flowchart of electromagnetic sensor KS operations 281
7.12 Interaction of electromagnetic sensor KSs and the blackboard 282
7.13 Illustrating AIRES data fusion 288
7.14 Electromagnetic sensor KBS implementation process 292
7.15 File-based data transfer 292
8.1 Crack classification 300

LIST OF TABLES

2.1	Inspection data recorded for a fixed platform	44
2.2	Example SCF predictions for all welded members	55
4.1	Comparison of fuzzy logic with classical logic	128
5.1	RISC System components	170
5.2	Schedule list structure	173
5.3	Software packages and languages reviewed	200
6.1	Tubular joint geometric data	214
6.2	Default data for modelling of repairs	217
6.3	Material properties	219
6.4	Loading modelled as a WASR	220
6.5	Loading modelled as a Weibull distribution	221
6.6	Data for the AVS SIF solution	222
6.7	Inspection POD data	224
6.8	Inspection POS data	225
6.9	Case study costs	226
6.10	Nodes and joints for the example structure	229
6.11	Sample analysis cases	229
6.12	Nordic Building Committee reliability index target levels	236
6.13	Reliability index target levels proposed in RISC	237
6.14	Reliability index target levels for the case study	237
6.15	List of actions for combined scheduling	240
6.16	Initial schedule of inspections	242
6.17	Schedule of inspections from depth first search	244
6.18	Schedule of inspections from breadth-first search	244
6.19	Schedule of inspections from best-first search	245
6.20	RISCREL input data	249
7.1	TRAM concept structure	267
7.2	TRAM knowledge source card structure	268
7.3	Example concepts	280
7.4	Example KS cards	282

1 INTRODUCTION

This thesis presents a solution to the problem of rational inspection planning for steel jacket structures in the offshore oil and gas industry. The approach is based on structural reliability analysis which is now known as *reliability-based inspection scheduling for fixed offshore platforms* (RISC), and is encapsulated in computer software with a knowledge base system architecture.

1.1 BACKGROUND

The application of reliability methods in the design and maintenance of offshore structures has been developed only relatively recently. For many reasons, such as the degradation of existing platforms and the requirement to be able to extend the life of some of these in order that satellites may be attached for the purposes of horizontal drilling, there is now a need to be able to analyse the reliability of such structures more rigorously.

One problem facing engineers is the maintenance of structures and in particular in carrying out inspections. Solving the decision-making problem of when, where and how to carry out inspection actions, involves using various tools. Traditionally, and in most industries, engineering judgement is employed to make decisions on maintenance strategies and inspection schedules for structures, and deterministic analyses are used to confirm these decisions.

In the offshore oil and gas industry in the UK, one of the most common forms of structure is the fixed steel jacket type of offshore platform. These are highly redundant and are subject to many random or uncertain factors, such as uncertainties in the load distribution through the components and in particular, to time-varying and cyclic loads and so to fatigue. Conventional reliability assessment of these structures is not always feasible, due to the complexity of the problem. Instead, use can be made of reliability analysis techniques recently developed specifically for complex structures, based on simplifications of the functions describing failure.

Another real problem lies in applying this analysis to the production of rational inspection strategies. Planning and scheduling requires understanding the inspection procedure and making use of the appropriate information on inspection techniques.

Finally, the difficulties in collecting input data for the analysis and of interpretation of the results remain. The interpreted results must then be combined to form a rational global solution for the structure which takes into account constraints which may not have been included, neither implicitly nor explicitly, into the analysis. Thus to provide rational IRM schedules, it is necessary to consider developing complete systems that will aid in carrying out analysis and interpreting the results in light of external information.

The problem of being able to process and use correctly all required data and information could be a major factor in limiting the use of what ought to be widely available information for improved scheduling of inspection for fixed offshore platforms. Some information is very recent and many models and analysis methods are relatively new. Therefore the analytical tools are not easy to use as there is little documentation and, more importantly, little expertise to exploit the information and the techniques within operator organisations. On the other hand, a platform may have been designed in an earlier time when more primitive information and methods were applied, and as a result not all required data on the structure may be available. Information overload is also a problem. Often there is too much data and information to be taken into account, and this alone can make it difficult for the operator to make full and rational use of it.

The development of a knowledge base system (KBS) for RISC provides a solution to the above issues. KBS technology was developed precisely in an attempt to provide a computer framework for the use of little documented and extensive information.

The work carried out for this thesis is the application of knowledge base systems to the problem of reliability-based inspection scheduling for fixed offshore structures. Such systems will enable operators to develop rational inspection, repair and maintenance (IRM) scheduling strategies.

1.2 HISTORY OF OFFSHORE STRUCTURES

The birth of the oil industry was in 1859, when the first drilling structure was erected in Titusville, Pennsylvania, USA. In 1896, the first offshore drilling occurred in Summerland, California: piers extending from the coastline provided the base for drilling to take place. A wooden platform was used in Lake Ferry, Louisiana, in 1909. The two first steel structures were installed in the Gulf of Mexico in 1947, in 5.5m and 7m of water. These were constructed completely offshore: small pilings of varying sizes and depths were driven into the seabed and on a construction barge, horizontal braces were cut to fit. By 1948, prefabricated sections of substructures were employed. This development was followed in the 1950s by the advent of onshore prefabrication of complete substructures, that is, templates or jackets, which were placed upright on to the seabed and piles were then driven through the legs into the seabed.

Gas was discovered at Groningen in Holland in 1959 and this marked the start of the search for gas and oil in the North Sea. The first offshore structure was installed in 1966 for gas production in the Viking Field, which was only 17 m deep. This was followed by the discovery of oil in 1969. The industry has been marked by rapid technological development and hence in 1974, installation of a platform in the much deeper Forties field (130 m) was carried out and was followed in 1975 by the start of large scale production of oil. By 1988, drilling started in the Troll Field in depths greater than 300 m. Over 100 steel structures were installed in the North Sea by 1989.

Fixed steel platforms are limited to a maximum water depth of approximately 400 m. Because of this limitation, other form of structures such as tension leg platforms and guyed towers have largely superceded them in recent years. Nevertheless, jacket platforms still represent a major economic investment which requires protection. In addition, in many fields, platforms are coming to the end of their design life but some are still required in order to exploit small outlying fields, and these structures may, in some cases, require some top-side modifications.

1.3 MOTIVATION

Jacket type steel offshore platforms can be very large and heavy structures. The jacket of a typical structure is the shell framework which supports the deck and topside modules for accommodation, drilling and production. A typical jacket structure is shown in Figure 1.1. The basic framework or jacket is fabricated from steel cylindrical members, where the horizontal members may be up to 1.5 m in diameter and the vertical members up to 3.5 m in diameter. The nodes (see Figure 1.2), or the intersections of the members, can be simple T configurations with one brace joined at right-angles to a main member or chord, or highly complex with many braces joined at the same point of a chord.

It has been found that the weakest part of these structures is the node points, that is, the welded connections between two members. Defects in the weld introduced during construction are difficult to identify and even when a weld can be said to be defect-free, it is in this area where cracks frequently appear. Hence it is at the welded connections at the nodes where much routine inspection for fatigue crack growth is aimed. The mechanism behind the growth of cracks and the problems in identifying defects are described in more detail in Chapter 2.

Underwater inspection is a dangerous and costly exercise. Resources, such as work force and time availability of equipment, are highly constrained. In particular, in areas such as the North Sea, weather conditions mean that diving can only take place during a short part of the summer. In addition, the planning of routine weld inspections is carried out against a background of guidelines and regulations dictating the proportion of the structure which must be inspected and the intervals between inspections and methods to be employed. Regulatory bodies, however, may allow offshore operators flexibility in their planning if a thorough technical justification for their schedules is given. Such a justification requires detailed analysis of the structure.

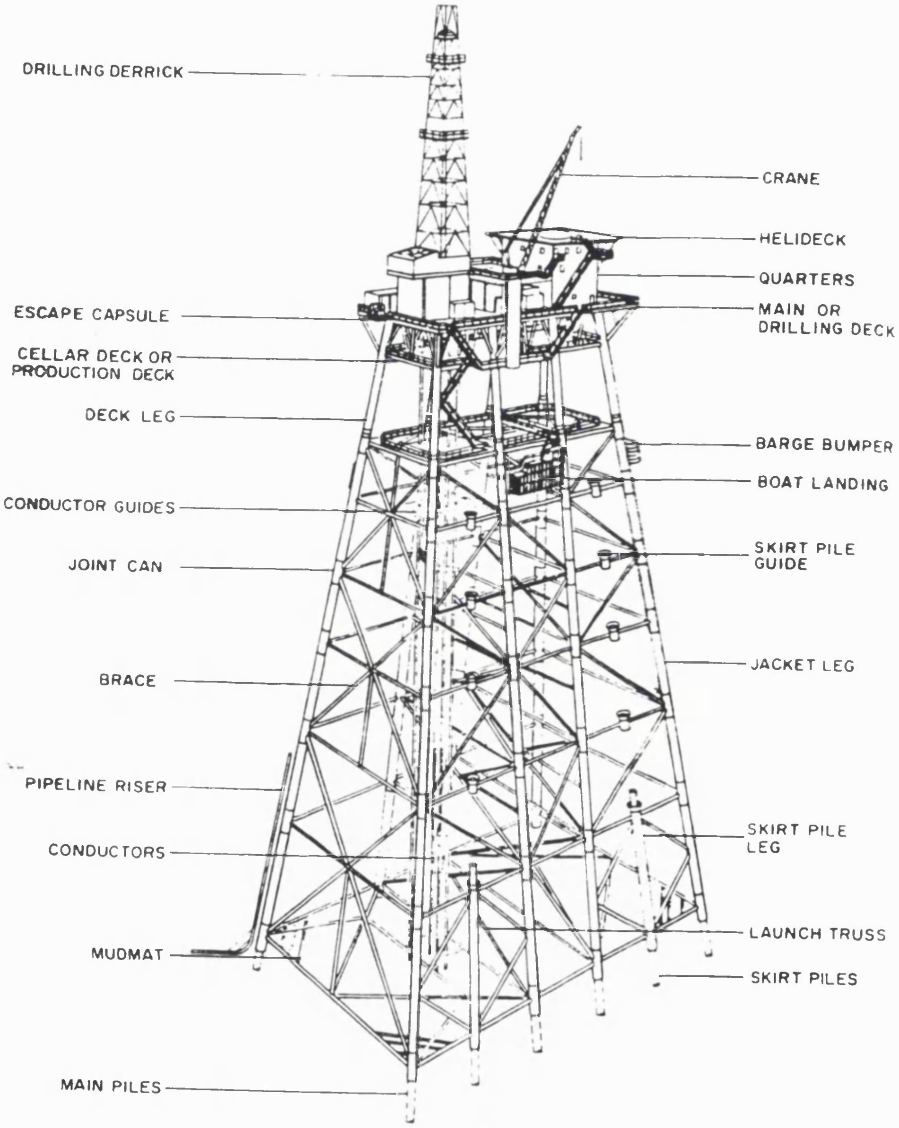


Figure 1.1 A typical jacket type platform (from Drawe and Reiffel, 1986)

A jacket platform is typically a highly redundant structure. This redundancy occurs for two reasons. During the manufacturing process, the structure lies on its side and hence members are required for 'horizontal' loads as well as for the final vertical dead loads. Additionally, redundancy is also introduced in following a fail-safe design approach, that is, should one member fail, the load may be redistributed to other members, until repair can take place. Although redundancy adds to the overall reliability of the structure, it also makes the structure more difficult to analyse.

Besides the sheer structural complexity, modelling realistically the loads applied to the structure is an added problem. Thus, the structural and stress analysis of nodes and joints can be complex. Detailed analyses based on the finite element method, for instance, are time-consuming and input models can be very difficult to define correctly. Analyses based on parametric equations, which encapsulate empirical knowledge of the behaviour of the node, are simpler to carry out, but the appropriate choice of equations must be made and the results carefully interpreted. Thus, the provision of a sufficient case for reduced or at least flexible schedules can introduce difficulties.

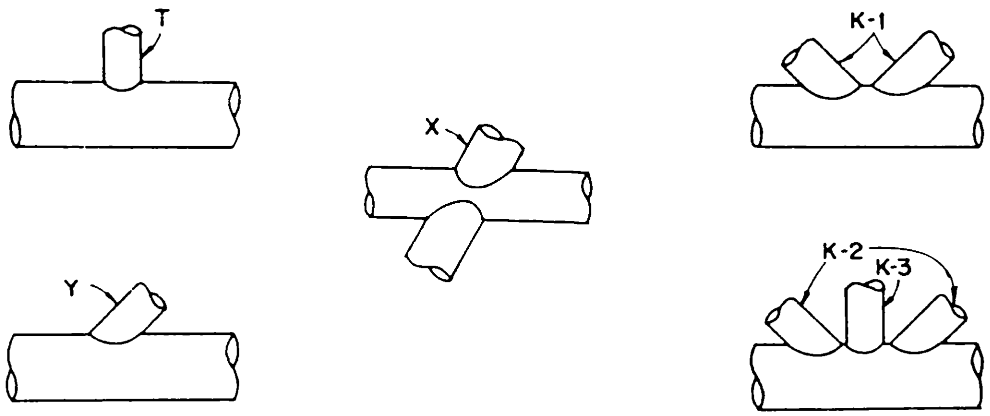


Figure 1.2 Typical node configurations for jacket type platforms

Finally, legislation and guidelines are being directed towards improved safety (Lang, 1990). There were few fatalities in the oil and gas industry until 1988: from 1976 to 1987 the average number of fatal accidents was of the order of 10 per year. In 1988, however, three accidents overshadowed the safety record of the industry. The Piper Alpha platform was destroyed by fire in July and 167 people were killed. The Ocean Odyssey rig was damaged also by fire in September, causing one death. A Shell oil-storage vessel broke free from its moorings in December. These accidents led to pressure on the government to apply higher safety requirements on operators.

The main motivating factor behind rational inspection and evaluation of structures is to confirm the continuing safety of the platform concerned, while not incurring excessive costs. A secondary factor, which is beginning to be ever more important, is the need to demonstrate the feasibility of life extension of jacket platforms.

1.4 GLOSSARY

The following table defines terms as used within this document. Italicised words refer to other terms in the table. Wherever a reference is made to a country, e.g. (UK), this is meant to show that the information is relevant to that country and may not be for other countries.

Action	A general term for any action (inspection, repair, etc.) to be carried out on the structure. Expected required resources (time, techniques, labour, etc.) are associated with each action.
Analysis module	Any external analysis software to be used by the RISC system. For example, COMREL (component reliability analysis), SYSREL (system reliability analysis), ULDAN (loading analysis software), FAFRAM and FACTS (two fatigue analysis crack growth programs), which were combined into one Component Fatigue Analysis (CFA) module, and the main program RISCREL, which is CFA, COMREL and SYSREL integrated into one reliability-based fatigue fracture mechanics analysis software for tubular joints. Analysis modules in the RISC system are regarded as *knowledge sources* and the creation of input files, running of the software and interpretation of output files are controlled by the RISC System.
As-installed data	The design data updated with the results from the first post-installation inspections.
Brace	A tubular member welded to another. It is cut to fit the geometry of the uncut member and other braces at the same intersection or *node*.
Chord	A tubular member on to which other tubular members, the *brace*s, are welded. It is the uncut member at the intersection point or *node*.
Concept	An item of knowledge relating to a class of objects, an object itself, or an abstract notion, stored in a *knowledge base. Knowledge sources* act on the data in a concept to provide more data for the same or other concept or to create new concepts.
Cost evaluation	Analysis carried out by the *analysis module* RISCREL taking as input a *maintenance plan* for a joint and costs associated with failure of the joint, inspection and repair to output expected costs.

Critical component	A non-redundant component (note that this term is sometimes used synonymously with *primary* q.v.).
Heuristic	Knowledge abstracted from experience rather than from formal analysis. Often also called a 'rule of thumb'.
Hot spot stress	The product of the nominal stress and stress concentration factor.
Inspection	A discrete monitoring action on a component at a point in time.
Inspection period	A period of time in a year during which *inspection* is carried out. Usually the same as the *weather window*, but in cases where some of the weather window period has been taken up by other activities, so restricting inspection, the inspection period may be continued beyond the weather window.
Interpretation	The interpretation of the results from any analysis module is carried out after they have been read into the system. A *rule set* type of *knowledge source* is employed for this process. The separation of the interpretation task from the reading-in of analysis results is essential to allow future changes made to the system due to new operator preferences or regulations.
Joint	The welded connection between any two members.
Knowledge base (KB)	A KB is a computer module containing stored knowledge in the form of *production rules, knowledge sources* or *concepts.*
Knowledge source (KS)	A KS is an independent and self-contained program which is controlled by a KBS control subsystem, or *system controller,* to carry out an identified task. A single *production rule*, a set of such rules or a complex software package can all be regarded as KSs. In the context of the RISC System, a KS represents either an *analysis module*, or a set of rules and procedures.
Knowledge base system (KBS)	A KBS is a software system that uses captured *heuristics* or other forms of knowledge in a *knowledge base* to perform or support tasks normally done by an expert or consultant. The RISC System is a KBS which provides intelligent support to the problem of inspection scheduling for fixed offshore structures.

Maintenance plan	A proposed *action* for a *joint* where the inspection technique, possible repair action and inspection time(s) are given.
Node	A set of welded connections between one *chord* and one or more *braces*.
Optimal scheduling	Optimal scheduling involves the combination of classical decision theory and modern reliability analysis methods only to find the least-cost schedule of *IRM actions* for a structure. It is usually only possible for a small number of actions.
PFI	Probability of false indications or the false call rate, that is, the probability of a nonexisting crack being detected by a given inspection technique.
POD	The probability of crack detection for an inspection technique, which is a measure of the technique's sensitivity.
POS	Reliability of sizing or probability of sizing for an inspection technique, which is a measure of the technique's accuracy.
Primary component	A component which has been pre-defined/agreed as being important for structural integrity.
Production rule	A knowledge structure with an 'IF-THEN' form. *Heuristics*, guidelines and regulations will most often be encoded as production rules.
Rational scheduling	Rational scheduling works on a large number of joints (c.f. *optimal scheduling*) using *heuristic* knowledge abstracted from operators' experience, management procedures etc. as the major problem solving strategies.
Resource deficit	A measure for an inspection period or schedule indicating the difference between the required resources and the available resources.
Rule set	A grouped set of *production rules* intended for a particular task.
Schedule	A list of *actions* to be carried out at each *inspection period* of the *scheduling period*. The list, also known as the Scope of Work, does not give the order of actions within inspection periods.

Scheduling IRM (inspection, repair and maintenance) model	The scheduling IRM model describes the processes and activities necessary to produce a valid schedule which satisfies time, resource and cost constraints. The model defines the tasks to be performed by a computer system supporting the scheduling procedure.
Scheduling period	The period of consideration for scheduling purposes: usually 5 years for the UK or 4 years for Norway.
Software system architecture	A software system architecture identifies the necessary components of the system to perform identified tasks. In the RISC System, this architecture refers to the core of the KBS which controls the inferences of *knowledge sources*, manages the interactions between system and users, and maintains the knowledge bases and databases.
Survey	A series of inspections and/or general review of the state of an area of the structure carried out over a single period of time.
Through-thickness crack	A crack that has grown through the wall of the member.
Weather window	Period of time in a year when it is considered safest for divers to carry out subsea activities.

1.5 NOMENCLATURE

The following table contains a list of mathematical symbols related to fatigue fracture mechanics and reliability analysis as used in this thesis.

$\underline{\alpha}$	The sensitivity vector, and OP $=\beta\underline{\alpha}$ where P is the design point, i.e. the point on the failure surface of shortest distance to the origin.
α,β,γ, and τ	Non-dimensional geometric parameters, used in the evaluation of the geometric SCF
β	Reliability index
β_C,β_{HL}	Reliability indices as defined by Cornell and by Hasofer and Lind
ΔK	The local stress range
μ_X and σ_X	The mean and standard deviation values of the random variable X

ρ_{XY}	A correlation coefficient between the random variables X and Y
σ_{HS}	The stress at the hot spot, that is, the maximum stress value for a welded connection
σ_n	The nominal stress
$\phi(.)$ and $\Phi(.)$	The standard normal density function and standard normal cumulative function
a	The crack size, which can be either be half of the length or the total depth according to the context
a_0 or a_i	The crack size at the start of the life of the component, that is, the initial crack size
a_{cr}	The critical crack size
$a(t)$	The instantaneous crack size due to fatigue at time t
a_T	Final or failure crack size
a_{NDI}	The detectable crack size for a non-destructive inspection technique
B	Magnetic field as a 3-dimensional matrix
$\mathbf{B}_x$	Component of magnetic field in x-axis stored as a matrix
$\mathbf{B}_y$	Component of magnetic field in y-axis stored as a matrix
$\mathbf{B}_z$	Component of magnetic field in z-axis stored as a matrix
c/a	The aspect ratio, that is, the ratio ½length:depth, of a crack
c_a, c_f, c_i, or c_r	Cost of an IRM action, failure, inspection or repair for a joint
C_{grind} and C_{weld}	Cost of grinding and welding repairs
$c(.)$	Cost function
C and m	Material properties for use with Paris' crack growth Law
Cov[X,Y]	The covariance between the random variables X and Y
da/dN	The instantaneous crack growth rate (m/s)
D	Total damage due to varying stress levels
ΔD	Damage due to one level of stress

E[X]	The expectation of random variable X or expected value of X, where $E[X]=\mu_X$ for the population of X; if considering a sample of X, E[X] is only an estimate of μ_X
$f_X(.)$ and $F_X(.)$	The probability density function (pdf) and corresponding cumulative distribution function (CDF) for a random variable X, hence $F_X(x)=P(X<x)= \int_{-\infty}^{x} f_X(x)dx$
$f_{X,Y}(x,y)$, $F_{X,Y}(x,y)$	The joint pdf and CDF for random variables X and Y
$f_{Y\|X}(y\|x)$, $F_{Y\|X}(y\|x)$	Conditional pdf and CDF for the random variable Y given variable X takes on value x
g(.)	Failure function or failure surface function or limit state function or state function, defined such that when g(.)<0 => failure, g(.) > 0 => safety, and g(.) = 0 is the failure surface
h(.)	The tangent surface to the failure surface in the normalised basic variable space
K	Stress intensity factor
$\underline{L}$	Geometric (lengths) basic variables
M	The safety margin, defined such that M<0 implies failure, and M =0 is the limit state
n	Number of stress cycles
n_i	Number of stress cycles at the stress level S_i
N	Number of stress cycles to failure
N_i	Number of stress cycles to failure for the stress level S_i
N_T	The number of stress cycles to the end of the design life
P	The design point, which is a point on the failure surface closest to the origin.
p(.)	Membership function for a fuzzy set
P(A)	The probability of an event A occurring
POD and POD(a)	Probability of Detection for an inspection technique, which depends on the crack size *a*

POF	The Probability of Failure = P(M<0)
POS and POS(a)	Probability of Sizing for an inspection technique, which may depend on crack size a
R and R(t)	Reliability at time t
$\underline{R}$	Resistance basic variables
S	Stress level
S and R	Single load and strength parameters
$\underline{S}$	Loading (stresses) basic variables
SCF_G	The geometric SCF
SCF_{HS}	The stress concentration factor at the hot spot
SCF_N	The SCF at the crack tip or notch, which is a function of $1/r$, where r is the radius of curvature of the crack at the notch tip
S_i	A particular stress level
T	Member wall thickness
T(.)	A transform operator
t_{insp}	Discrete points in time corresponding to inspection periods
$t_{insp}*$	Point in time corresponding to the current inspection period
T_L	Expected lifetime of a structure
$\underline{U}$	The normalised basic variables
u(.)	Utility function
$\underline{X}$	A vector representing the basic variables
$\underline{Z}$	Standardised basic variables

1.6 ORGANISATION OF THE THESIS

The aim of this thesis is to present the problem to be solved, followed by a description of the analytical tools and approaches which can be applied to solve the specified problem and finally a proposed system for IRM scheduling.

In Chapter 2, the current procedures for IRM in the offshore industry are reviewed. In addition, as crack growth due to fatigue is one of the main causes of degradation of steel jacket platforms, an explanation of the basic principles of fatigue fracture mechanics analysis is given in the second half of the chapter. The purpose of this chapter is to assemble the current state of IRM scheduling and the basic analysis tool, that is, fatigue fracture mechanics, for estimating the life of a structure. The information gathered for this chapter formed the basis for the requirements specification for the RISC System and provided the information required to formulate the scheduling model for RISC.

The fundamental concepts of structural reliability analysis are explained in Chapter 3. The required decision theory, the modelling of uncertainties for offshore structures and the analysis software RISCREL and ULDAN are also described. This chapter is a review of the current state-of-the-art in structural reliability and explains the Level II reliability analysis techniques First and Second Order Reliability Methods (FORM and SORM). Some early assumptions made for this work and necessary to be able to apply structural reliability analysis to IRM scheduling were based on the requirements specifications produced by the author. The specifications were also used to define the interfaces to the reliability software package RISCREL, for the FORM and SORM analysis and cost evaluation of IRM actions and which was implemented by other colleagues during the RISC project.

A review of applied artificial intelligence and knowledge base systems is given in Chapter 4. The formalisms and applications described provide a basis for defining the architecture for a knowledge-based IRM scheduling system and the algorithms employed. The discussion of the implementation issues provides an understanding of the choices made in the implementation and management of the IRM scheduling KBS. Constraints-based scheduling based on basic searching techniques and concepts from constraints satisfaction is explained in some detail here.

After a thorough study of the IRM problem and reviews of the tools that would provide a solution to the IRM problem, work was carried out on a computer-aided RISC System. The design and implementation of this IRM scheduling KBS is described in Chapter 5. Details are given of the Scheduling Model, that is of the procedures employed to carry out rational inspection planning based on the analysis results. Details of the design are described. A software review was carried out of the possible development tools and a condensed version is given here. The implemented RISC Demonstrator provides the basis for the final RISC System.

A case study is presented in Chapter 6 to illustrate the methods and the use of the complete RISC System. The data for the study was gathered from a variety of sources, including example structures provided by several sponsors of the RISC project. The complete procedure is demonstrated, including the generation of possible maintenance plans and the input to the RISCREL software, the interpretation of the results from RISCREL and the algorithms for constraints-based searching for the scheduling of the inspection actions.

One issue, which arose early in during work for this thesis, is the problem of interpreting inspection results. In Chapter 7, work carried out as part of the AIRES project to develop a prototype automated inspection system that provides intelligent interpretation of electromagnetic sensor data is presented. In this chapter, some implementation problems, particularly regarding knowledge representation and data fusion, are discussed.

Chapter 8 describes the future work which may be carried out to provide a scheme for full systems reliability-based design and IRM scheduling for structures. This includes a description of how automated inspection interpretation systems such as AIRES may be integrated into RISC. Also considered are extensions to the RISC methodology to consider, for instance, systems reliability and case-based reasoning. Finally, the extent to which RISC may be applied to other structures and in other industries is discussed.

1.7 Summary of Work

The work carried out for this thesis was carried out as part of two projects:

- **Reliability-based Inspection SCheduling for fixed offshore platforms (RISC)**

The RISC project was approved by the Commission of European Communities for partial funding under the THERMIE programme. The leading partner in this project was Technical Software Consultants Ltd (UK); other partners were University College London (UK), Registro Navale Italiano (I), University of Pisa (I) and TNO-IBBC (NL). The total budget was set at £1,222,000 of which £455,164 was funded by the EC and £221,000 was non-UK funding. Some of the reliability analysis software modules developed within another EEC-funded project (BRITE P2124) were included and further extended with collaboration with RCP ApS (Denmark) and RCP GMBh (Germany). The RISC project ran from the 1st May 1991 to April 1994. The deliverable from this work was the RISC System demonstrator: a knowledge base system with analysis modules and sample databases integrated to provide inspection schedules.

The author's work in this project was primarily on the development of the knowledge base system. This included involvement in the early definition of the problem; responsibility for the requirements specification; and a major role in the definition of the scheduling model. A functional specification for the complete system was provided by the author. The programming of the demonstrator system was carried out under the author's supervision from design documents produced by the author.

- **Automated Image Reconstruction using Expert Systems (AIRES)**

The aim of the Automated Image Recognition using Expert Systems project was to provide the technology for automated total surface inspection. The inspection of components or materials was to be carried out during both production and service. Interpretation of the data from the sensors was to be carried out within an expert system framework for image reconstruction from multiple sensors. For this, an integrated software and multi-sensor system was developed, based on a CCD camera providing vision data and on an electromagnetic sensor employing the alternating current field measurement (ACFM) system. The total budget for the 2-year project was £1,814,000 partly funded by the CEC under the BRITE/EURAM programme. The partners were Technical Software Consultants Ltd (UK), University College London (UK), Commissariat a l'Energie Atomique (F), University of Hanover (D), Peugeot SA (F) and Deutsche Aerospace SA (D).

The author was responsible for the work carried out for the electromagnetic sensor KBS. This involved understanding the use of the ACFM inspection technique, and also of the KBS development tool developed by LETI; overseeing the methods undertaken for knowledge acquisition from the ACFM experts; and specifying the data structures required for the interpretation of the ACFM data.

1.8 SUMMARY

The widespread use of reliability-based inspection scheduling systems in a knowledge-based framework will help to improve the fundamental understanding of the problem. In addition, the use of such systems may lead to improved overall reliability of offshore structures without incurring economic penalties.

The basic approach described in this thesis could be applied to other forms of large structures in other industries, where inspections of a similar nature have to be carried out *in situ* on similar components. Examples of these include pipelines, steel bridges, production plants and aircraft.

2 INSPECTION, REPAIR AND MAINTENANCE OF OFFSHORE STRUCTURES

Inspection and other monitoring actions are carried out on structures to find out if any damage or deterioration has occurred. Any damage found requires evaluation to decide upon future maintenance actions. The initial problem is to decide on a plan for inspection.

For offshore structures, fatigue crack growth is one of the main deterioration processes, particularly in deep waters. Other causes of deterioration are corrosion leading to loss of section, marine growth leading to increased loading and impact damage due to dropped objects or ships colliding. Nevertheless, the fatigue life of welded connections in offshore structures is a governing factor for the planning of inspection and maintenance actions, especially in areas such as the North Sea, where weather conditions are unfavourable. The occurrence of frequent storms leads to a large number of loading cycles. Also, deeper waters imply taller structures. These are prone to a greater dynamic response because ensuring that the structure's natural frequency is very different to the loading frequency becomes more difficult for taller structures, which means greater fatigue problems. High stress concentrations at welded intersections lead to fatigue cracks growing from weld defects acting as crack initiators.

Since many uncertainties are inherent in the analysis of the deterioration process of offshore structures, the use of reliability techniques, which will consider uncertainties, combined with fatigue fracture mechanics analysis is being considered. Further, operators are moving towards adopting a reliability-based approach to deciding where and when to inspect offshore structures and in particular the tubular welded-connections of steel jacket platforms. This approach is endorsed by certification authorities, as these methods provide a more rational basis for decision-making (Birkinshaw et al, 1993; RBI, 1989; DnV, 1982).

To provide the requirements for a computer-aided system to aid inspection, repair and maintenance (IRM), a review of existing practice for IRM of offshore platforms in the North Sea was carried out. A Working Group of representatives from the offshore industry was consulted and provided further insight into existing practices in maintenance and inspection scheduling. Interviewing the experts and operators followed the recommendations in the Research Task Force on Risk-based Inspection Guidelines report (RBI Research Task Force, 1989). Additionally, literature reviews were carried out and in particular, a report from The Marine Technology Directorate on Underwater Inspection of Steel Offshore Structures provided much background information (MTD, 1989).

The Reliability based Inspection SCheduling (RISC) methodology is based on fatigue fracture mechanics as a model of the failure mechanism, combined with reliability analysis to take into account uncertainties. Reliability-based fatigue fracture mechanics analysis is then integrated into a rational scheduling system that incorporates current practice and guidelines. The main aim of the RISC methodology is to enable useful IRM planning, based on objective, quantitative analyses which estimate the expected costs of IRM actions, but allowing operators' preferences and requirements to take precedence if necessary.

This chapter describes the inspection, repair and maintenance (IRM) activities carried out on fixed offshore platforms and provides the background for the requirements for a computer system, the RISC System, to aid IRM planning based on the RISC methodology. The final section of this chapter briefly reviews current approaches for maintenance planning, including consideration of uncertain or random factors for offshore structures.

2.1 AN OVERVIEW OF CURRENT PROCEDURES

This section describes the current procedures carried out by operators for in-service inspection, repair and maintenance of an offshore fixed platform. Only the problems relating to the scheduling of inspection and repair of the substructure are discussed. Inspection of components in the superstructure or deck is not considered and the actual techniques employed are only outlined.

2.1.1 Guidelines and Legislation

An IRM system needs to take into account current and possible future legislation and guidelines for the production of usable schedules. Current statutory requirements in the UK are that the certifying authorities issue a Certificate of Fitness for each platform. This certificate is valid for five years, or four years only for Norwegian waters based on the requirements laid down in The Offshore Installations (Construction and Survey) Regulations (1974) (MTD, 1989). Furthermore, the installation owner or operator has to demonstrate the integrity of the structure to receive the certificate.

The starting point of the procedure to demonstrate integrity of a structure involves the operator and certifying authorities or agencies agreeing on a set of *primary* nodes or components, based on criteria that are discussed later. The operator is required to inspect these primary components over a period of four to five years usually. Regulations state that these inspections must be carried out at the end of one certified period to obtain the next certificate. A programme of inspections carried out over the life of the certificate is usually allowed, however, if the schedule is approved by the certifying authorities. Approval is dependent on the operator ensuring that all primary joints are included.

Figure 2.1 represents the current procedure as laid down by the Department of Energy (DEn, 1984) where a joint is inspected at quarter life intervals, that is, every five years, if the design life is twenty years.

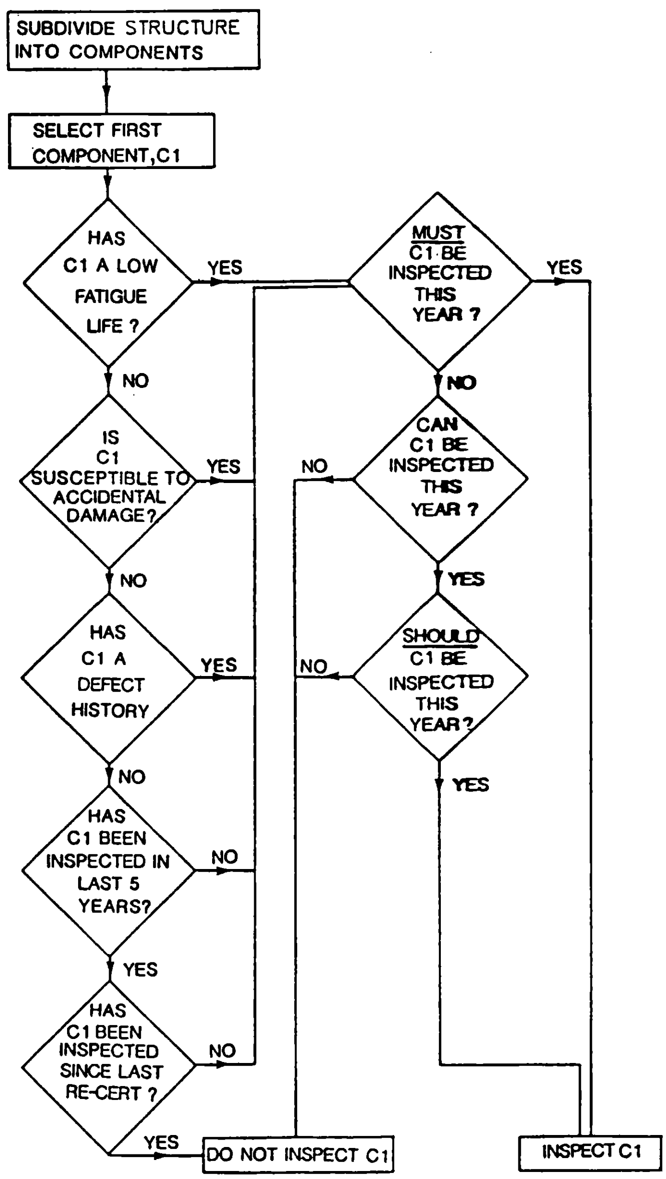


Figure 2.1 Current IRM procedure described by the Department of Energy

In certain countries, no definitive regulations exist. Instead, certification authorities require justification from the operators that the analysis carried out to produce a plan of inspections is sufficient. As an example, in Norway, operators are largely self-certifying, although Det norske Veritas guidelines include the need to inspect all primary nodes at least once every five years (DnV, 1982).

To summarise, current representative regulations/guidelines include:

- a five-year period, in the UK, or four years in Norwegian waters, in which to inspect all primary nodes
- 2nd annual inspection, in the UK, of all damaged joints
- incident inspections to ensure integrity of structure
- annual surveys may be implemented instead of one major survey covering all the structure

Current practices incorporate guidelines relating to the definition of primary nodes, the extent and methods of analysis, and the interpretation of the analysis results.

2.1.2 Participants

Several organisations are involved in the production of an inspection schedule and so information from each participants is required for an IRM computer-aided system. During each phase of the inspection scheduling process, each participant carries out a different role:

1. The operator, or a consultancy to the operator, produces a schedule.
2. The operator proposes a schedule to the certification agency.
3. The certification agency agrees to the schedule or requires some modifications or justification for the schedule.
4. A subcontracted inspection company carries out inspection tasks according to the schedule.
5. Repair specialists may carry out any major repair required.

The designers may not be involved with the operation of the structure and therefore do not necessarily design with maintenance in mind. In Norway, the operator's maintenance department will provide a list of tasks and of joints to be inspected to its sub sea operations department, which in turn will attempt to produce a schedule for the list. The time from producing a schedule to carrying out the inspection may be up to 6-9 months. This lag may be significant as the schedule may be produced with out-of-date information on the structure. After inspection, re-analysis of the structure may take place to decide the repair actions or if re-inspection needs to take place.

2.1.3 Selection Criteria

As inspecting all parts of the structure frequently is impossible, design and analysis data is taken into account to identify important or primary nodes. A node or any component is defined to be primary if it is agreed by the certification authority and the operator to be important in some way for structural integrity. This may be because the node is a non-redundant component, it has been assessed to have a low fatigue life, it is damaged and requires frequent monitoring to ensure safety or a combination of these factors. Inspection effort is then concentrated on these primary nodes with parts of the platform known to be at risk of damage, for instance due to collisions.

For a jacket structure, such as can be found on shallow waters in the Gulf of Mexico, the number of nodes may be as little as 300. In the North Sea, for a typical structure the total number of nodes is approximately 1500-2000. The maximum number expected for any structure is up to 3000 nodes. The number of agreed and predefined primary nodes can be approximately 200 to 300.

Every year inspection is usually carried out on:

- a percentage of the primary joints
- nodes with known weld-toe cracks or defects, found in past inspections
- members and nodes that are possibly damaged due to known incidents
- past repairs (including grinding carried out at inspection)
- areas of severe marine growth which will require cleaning

In the UK, the data on damaged nodes and members are stored in the damage register of the structure, and all components included in the register are inspected regularly.

In order to rank the joints, current guidelines suggest associating a weighting to each joint j given by the following:

$$\text{Weighting} = (Yj \times Xj) + Wj \qquad (2.1)$$

where Yj is a weighting representing the consequence of failure, Xj represents the likelihood of failure and Wj is a factor that is dependent on the number of years since the last inspection. The consequence of failure is based on a set of ten factors including redundancy and risk to life; the likelihood of failure is the sum of weighting for each type of failure mode; both include a factor representing confidence in the assessment. The details and recommended values for the weighting that may be assigned to each type of member can be found in the Marine Technology Directorate report for Underwater Inspection of Steel Offshore Structures (MTD, 1989). The higher the total weighting, the more likely it is that the node requires inspection.

Values of the Yj and Xj factors range from 10 to 450 and 15 to 250 respectively. Wj values step in value, nonlinearly, from 0 for inspection in the previous year, up to a maximum 6400 for inspection more than 5 years before. As an example, consider an 'average' node inspected three years before. The medium ratings of both X and Y are 80 and the W factor is 2560. So the total weighting for this joint is 80 x 80 +2560 = 8960. For a very safe node, with a minimum likelihood of failure and a minimum consequence of failure, inspected more than 5 years ago, the total weighting is 15 x 10 + 6400 = 6550. In contrast, for a highly critical with a very high likelihood of failure and inspected last year, the total weighting becomes 450 x 250 + 0 = 112500. Normally it is not expected that any one node would achieve such high X and Y factor values.

Figure 2.2 shows the DEn procedure that makes use of weighting factors in an attempt to incorporate reliability into IRM planning.

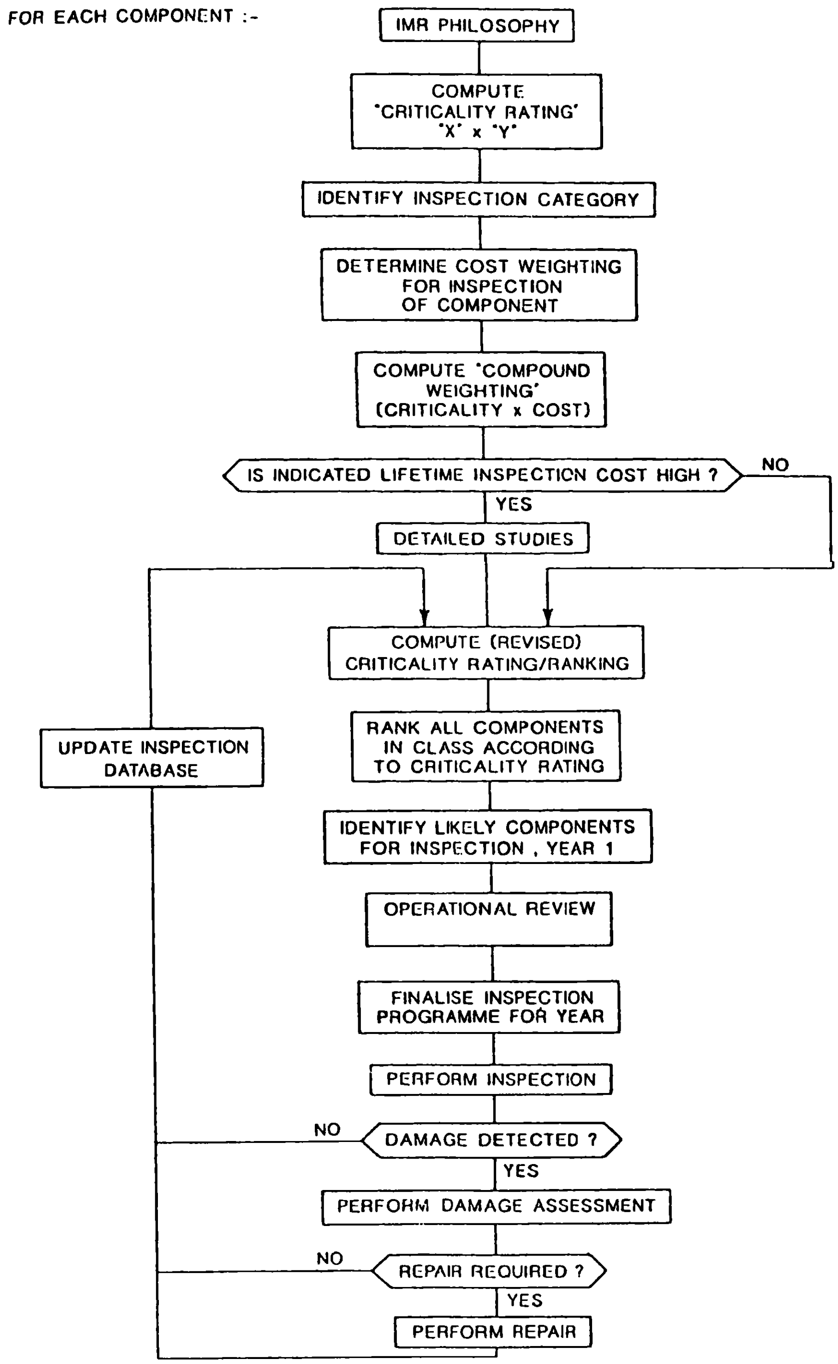


Figure 2.2 DEn IRM procedure making use of weighting factors

For certain components, inspection may be carried out once only and post-installation to ensure the integrity of the structure, and to confirm that no damage has occurred during installation. An example is the possibility of damage to the principal legs during post-installation removal of the wooden runways, as found in some Mexican structures (IMP, 1992b). Once confirmed that no such damage has occurred it may be safely assumed that the principal legs in place are as designed. Usually, post-installation inspection can be integrated as part of the first inspection schedule, but is used as part of the as-installed data.

2.1.4 Analysis Methods

Detailed analysis of a platform is carried out at the design stage, and when

- damage is detected, such as dents from impact, or extensive cracks
- changes are made to the structure, such as extensive repair or when new loads are added on to the deck, or changes are made to the layout of the deck, etc.
- codes or guidelines change, to ensure that the structure complies with new regulations

In the UK until the mid-1990s, the analysis of an offshore oil platform was essentially deterministic, that is, the assumptions were that the structural properties are time invariant , the frequency of inspections fixed, and the results of inspections are accurate. In contrast, probabilistic and reliability analysis has been allowed by the certification authorities in Norway for some time now as justification for a proposed schedule.

From the information provided by the Working Group, current analysis procedures together with the role they each play in inspection scheduling were identified:

- Static strength analysis involves little analysis, and is employed to identify components that are likely to fail.
- Redundancy analysis which simulates the failure of members or nodes, is carried out in the UK and requires the use of data-files held by the designer organisation. In Norway, the SRS model, or Structural Re-analysis System, is used. Both techniques identify critical components.
- Collapse analysis is carried out in the UK, and Push-over analysis in Norway. These analyses are carried out in the nonlinear range, implementing codes such as USFOS, to predict the consequence of failure of a component.
- Appropriate S/N curves can be used to calculate the expected life of a component. Designers use conservative stress concentration factor parametric equations and thus the

results show shorter fatigue lives than may be the actual case (Nicholson, 1986). Components may be ranked according to their fatigue life based on an S/N curve.

- Fracture mechanics or crack growth analysis predicts the remaining life of a joint for a given initial crack size.

2.1.5 Scheduling Constraints

The main scheduling constraint is that whenever possible inspection is carried out within a weather window, in Norway for instance, this window lasts for 50-60 days in the summer. Production operations on the structure, such as drilling, regardless usually take priority. This often affects the inspection procedure in the UK, in that inspection may be kept outside the weather window because of other operations taking precedence.

As the RISC methodology includes allowing the expected costs as an objective decision-making parameter, the costs of carrying out actions, and of the failure of the structure are also considered to be scheduling constraints.

The actual operation of inspections is constrained by many economic considerations:

- each inspection technique has associated with it an approximate cost per node, which may restrict the choice of inspection technique
- the costs of employing more than one vessel from which the inspection operations are carried out are prohibitive: approximately £50,000, per day in 1992 in Norway
- the total cost of moving the inspection vessel around the structure is typically of the order of £50,000, according to UK prices in 1992

The last two considerations introduce other constraints that affect the way in which diving operations are carried out:

- the position of the diving vessel leads to geographical constraints: the part of the platform that may be inspected is the area closest to the diving vessel
- there is a restriction on the total number of divers available, since only a few divers, usually two, can operate from the same vessel (UK, Norway)

Other economic concerns must also be taken into account. There is a trade-off between the employment of high-performance and efficient inspection teams and vessels, and the use of less expensive crews of divers. The former is associated with a short season for inspections and the increased sensitivity to bad weather spells and breakdowns, and therefore the required higher complement of offshore management to tackle any problems; the reduced cost of the latter allows these divers to be employed over a longer period and means that the schedules are not so susceptible to disruptions (Christer et al, 1989; Rivers, 1986). For an inspection scheduling decision support system such as the RISC System, it would be assumed that such economic considerations are made outside the system. The duration of the weather window and certain inspection procedural details, such as length of diving shifts, would be predefined.

Other constraints that may be included are on the use of remotely-operated-vehicles (ROVs) and those taking human safety into account:

- ROV inspection is carried out only for nodes that are deeper than 10m from the sea surface
- diver inspection from air-tanks takes place between the water level and 25m below the surface
- saturated diving takes place below 25m from the sea surface

It should be noted here that accurate cost information could be important for the application of RISC. Nonetheless, actual costs need not be provided since information on the relative costs will be sufficient, as the main purpose of identifying the costs is to provide a measure of 'undesirability'.

The size of the diving operation is considerable. In a survey carried out in 1985 it was reported that on one day in July, the peak period for the North Sea, 306 diving supervisors, 718 air divers, 680 mixed gas and bell divers and 167 life-support divers were active: a total 1871 diving personnel. It is estimated that approximately twice this number were working in one day in 1985 (Christer et al, 1989). In addition, to support one diver requires the backup of 50 surface personnel, at a weekly cost of £250,000 (1989 figures). To inspect a large structure in its entirety requires approximately four calender years.

Finally, a cost is associated with the fatality rates of divers. This cost cannot be taken into account directly here, but it does allow the consideration that if the numbers of inspections are reduced then the number of diving accidents may be reduced. The probability of fewer accidents due to fewer inspection tasks may be traded off against the higher probability of failure for components and ultimately of the structure.

2.2 INSPECTION OF TUBULAR JOINTS

The main purpose of inspection is to assess the current state of the structure and in particular of the component (Bray & Stanley, 1996). The presence of surface-breaking cracks indicates that fatigue damage is occurring in the node. The depth of the crack shows the severity of the damage, while the rate of crack growth provides a predictor of the remaining life of the joint. It is generally assumed that if a crack has grown through the thickness of a joint, then the joint has failed and immediate repair is required. Thus, one requirement of inspection on tubular joints is to identify fatigue cracks and to measure, in one way or another, the size of the crack.

Inspection techniques used for non-destructive evaluation(NDE) are either active, in that some probe or other element is applied to the component and an immediate response is measured, or passive, where monitoring is carried out over some reference period to check for some secondary reaction indicating a problem(Bray & Stanley, 1996).

This section describes the inspection procedures carried out to detect surface-breaking cracks on tubular joints in particular, which for steel jacket platforms active techniques are more often employed.

2.2.1 Inspection Techniques

Various inspection techniques for crack detection and sizing are currently in use in the detailed inspection of the sub-deck tubular joints. A summary follows of the main techniques reported in detail in the MTD report (1989) and in use as of 1992.

2.2.1.1 <u>Visual Inspection</u>

Close visual inspection (CV) is the most common technique. The diver cleans the weld and inspects it visually, or takes photographs or a video for later interpretation. The effectiveness of close visual inspection is significantly affected by the underwater environment. Thus, although it can be very rapid and cheap to implement, if the conditions are bad or the marine growth is heavy requiring much cleaning, it is often inefficient compared with other techniques. The method is used for detection of defects only: if a defect is seen, then another method is often used to size it.

2.2.1.2 <u>Magnetic Particle Inspection</u>

MPI is the next most common technique, employed by more than 80% of North Sea operators. It is most often used for detection, but it is sometimes used for sizing, although it only gives length measurements. The accuracy of the length measurements is undoubtedly not high, but it does at least detect reasonably well.

After detection with MPI, confirmation of the defect is usually carried out by either using another method for sizing, such as ACPD, discussed below, or carrying out *proof grinding*. In proof grinding, the diver removes by grinding a thin layer of the surface and reapplies MPI. If no crack is then detected, it is considered to have been removed and the smoothing of the weld surface implies an improved fatigue strength of the joint. If a crack is still there after grinding to a usual maximum of 2 mm, then further investigation of the crack and the joint is required.

2.2.1.3 Eddy Current

Many commercial systems are based on eddy current methods which give length, though not depth, measurement. It is a well-established technique for land structures, but it is still considered expensive and difficult to use on offshore structures. The diver needs only to move a hand unit over the weld of the joint, but the surface operator is required to be highly trained to interpret the sensor data.

2.2.1.4 Ultrasonic Characterisation

Techniques based on ultrasonics are contacting methods giving depth measurements. Ultrasonics characterisation is highly sensitive to operator performance and the equipment characteristics, and the defect's position and shape. In addition, the surface of the joint requires careful cleaning to remove all roughness. The difficulties in sizing cracks within an acceptable accuracy underwater means that ultrasonics is used less than other techniques in the offshore industry, although some work is being carried out in solving these problems with the time-of-flight diffraction (TOFD) technique based on ultrasonics.

2.2.1.5 Alternating Current Potential Drop and AC Field Measurement

These are relatively new methods developed at UCL, which will give depth measurements and therefore are ideal techniques for sizing of defects (Collins, 1995). ACPD is a contacting technique that measures the potential drop across a metallic surface due to the presence of a crack. The newer ACFM method is non-contacting and measures the changes in the electromagnetic field above the surface again due to the presence of a crack or pit.

ACPD or ACFM can be used with MPI, where the latter technique is best employed to detect cracks and the former is used to size cracks.

2.2.1.6 Flooded Member Detection

Other methods of detecting fatigue cracks include flooded member detection (FMD), which can be carried out by use of various techniques. FMD is a passive technique in that the check carried out is for the presence of a leak which in turn indicates a through-crack. FMD must be treated carefully: if failure is defined as occurring when a crack has penetrated the wall of the member, then a positive indication of FMD represents failure of the component.

FMD will certainly allow an operator to detect failed joints, but does not allow the progressive measurement of cracks as they grow or develop. As it is necessary to be able to detect cracks before failure to avoid failure of the joint, FMD can only be used as a backup method, and not for planning purposes.

2.2.2 Inspection Data

Much inspection data is currently collected and there are clearly no standard forms, or even types of data, which are common across operator organisations. In the case of data on cracks found at the weld toe of welded joints, information currently recorded on a crack will include

- indication length and, if possible, depth
- clock position given relative to some predefined point on the joint
- grinding length, width and depth

Other features of a weld for which information is recorded are

- corrosion and pitting in a qualitative form
- undercut in quantitative measures
- grinding marks as descriptive, qualitative information

All the inspection information is stored in various formats and in different locations:

- diver or inspection job-sheets
- engineering assessment reports
- platform damage status register

Most data is held as text and in tabular form. In addition, information on cracks and other defects found both at component level and at structure level (for example, from swim-round surveys) may be kept in a pictorial form. Such pictures may be drawings held as hand-drawn diagrams, computer graphics files, e.g. AUTOCAD in the UK, and photographic evidence, as both stills and in video format.

From the information provided by the Working Group and the above, the types of inspection data collected every year can be listed and are described in Table 2.1 overleaf.

Some data is already kept in computer databases and operators are making use of database technology, wherever possible, for storage of inspection results. For example, the Instituto Mexicano del Petroleo store data in a specialised inspection database (IMP, 1996). Operators in the Norwegian North Sea make use of DnV's PIA (Probabilistic Inspection Analysis) system that includes a central database of past inspections (Vardel & Moan, 1997). More advanced examples of computers to store data include the possible use of expert systems that provide lists of joints to be inspected and of tasks for each year to aid scheduling in Norway and in the UK (Khong & Lucia, 1991; Ahmad et al, 1993).

Table 2.1 Inspection data recorded for a fixed platform

Inspection Data	Type	Coverage
swim-round survey for general damage and marine growth	inventory	100%
flooded member detection providing information on the existence of through-thickness cracks	yes/no	100%
seabed debris survey to check that no items or objects have dropped onto the seabed, either from the structure itself due to unnoticed impacts or from the top-deck, and may have caused damage on falling down	inventory	at four locations
scour survey to check integrity of foundations	quantitative	
cathodic protection measurements	numeric	one leg
anode condition given as subjective information either as a grade or as an estimated percentage of eroded or remaining volume	graded (~3-4 grades)	one leg
marine growth information on depth of growth for cleaning	numeric	
bolted connections' information on the condition of the connection, such as cleaning and re-tightening required, visually inspected every year	qualitative	100%
welded joints data on cracks and other defects and immediate repair information in formats highly dependent on the type of inspection technique used and on the operators' preferences	quantitative /qualitative	selection of joints

2.2.3 Inspection Procedures

Every year, monitoring tasks involving inspections of individual components and surveys of complete sections of the platform are carried out. Inspections are made of some primary nodes, all known/expected defects, previous repairs and areas with severe marine growth. General surveying is carried out to obtain information on marine growth and possible or known incidents. The thickness of marine growth is measured because marine growth increases the effective size of the tubular members considerably, which in turn affects the loading on the structure. After incidents, the structure will be surveyed to ascertain if any major impact damage has occurred.

The main current areas of concern are that of the individual inspection actions. The decision-making process for scheduling of inspections does not need to consider surveys, as these require the use of a nearly constant level of resources. For RISC, the assumption has been made that only discrete monitoring activities listed above as inspection tasks are an issue and need to be considered in detail for rational planning and scheduling.

2.2.4 Other Tasks

The main aim of inspection and surveying is to check for damage or potential damage to the structure. At the time of inspection, several other tasks may also be carried out simultaneously on the component, such as immediate cleaning or minor repair. The act of inspection itself may require cleaning, but the main reason for specifying cleaning is to remove marine growth. Marine growth thickness measurements can be included in some way in the loading analysis and when marine growth is considered to have increased the loading appreciably, it is removed. The task of cleaning members is scheduled in to the overall maintenance plan.

In the UK, inspectors are permitted to carry out confirmatory or proof grinding, usually up to 2mm depth. Proof grinding carries out the double role of confirming the existence of a crack and then, in the case of a shallow crack, removing it. Further grinding repairs are also often carried out up to a permitted maximum percentage of chord wall, after which the weld is re-inspected to confirm removal of defects. For instance, 6-8mm deep and 200 mm long cracks have been removed by immediate grinding.

In Norway, any anomaly or detected defect is reported by the inspecting diver and the topside engineer to the maintenance department. As 24-hour reporting exists, a report will be followed by immediate analysis by the maintenance department to confirm the existence of a crack or defect and to classify it. Analysis is also used to decide whether immediate repair is required or if postponing the repair until the following weather window is safe, although the usual practice is to carry out immediate repairs.

Minor repair, in the form of grinding, may be carried out by the inspecting diver. More extensive repair actions are passed over to specialists to carry out. If a repair is to be carried out by participants other than the inspection organisation, then the schedule of future inspections will be unaffected. Major repairs may be carried out by welding or even clamping. Both affect the behaviour of the node. Welding changes the material properties and the initial crack distribution in the joint, while clamping requires extensive re-modelling and re-analysis of the joint.

2.2.5 Summary

The sequence of events in the inspection of a welded joint may be summarised as follows:

1. an inspection is performed
2. the inspection results are interpreted to give inspection data
3. on discovery of an anomaly, confirmation is carried out, possibly by use of proof grinding
4. if the defect or crack is confirmed, then it is reported and analysed
5. if appropriate, immediate minor repair, in the form of further grinding, is carried out by the inspecting diver and is reported

A computer system for IRM planning would be required to store and handle information related to interpreted inspection data, any confirmed cracks or defects and the repairs carried out. The way inspection information is stored should reflect the way it is recorded, that is, the database structures or file formats should allow the input and storage of data in the same form as when first recorded on paper. Such a computer storage system would allow large amounts of data to be analysed as part of trend monitoring, during audits and to enable historical records to be built.

2.2.6 Operators' Future Requirements

One way in which the RISC methodology and other work in this area may affect future requirements is that regulations and guidelines will be modified. For instance, the UK guidelines for defining primary joints may change, affecting the number of joints to be inspected once every five years. Furthermore, it is the current accepted practice that repairs always take place when a crack or defect is found (Topp, 1985). This policy may be found to be not required. Further, it may compromise the integrity of the structure which would be contrary to recent recommendations from the Health and Safety Executive on using the As Low As Reasonably Practicable (ALARP) principle (HSE, 1992; Sharp et al, 1993).

A particular change may be in the level of inspection effort required, as it may be found that carrying out inspections every year may no longer be necessary. This would then affect the assumptions made in producing a schedule and in fact true optimisation of the schedule may be carried out.

As reliability analysis techniques become more accepted, more data may be required on the amount and form of the reliability analysis method carried out. More up-to-date loading and other environmental information may be recorded. The incorporation of human factors, either explicitly or at least in the modelling of random variables, will be required. Further analysis may be carried out on the collected data to update uncertainties and parameters, for instance measure variances in inspection performances, of both the equipment and the human inspectors (Rivers, 1986).

Inspection equipment and how it is used will also change with time. Very recent work at University College London for the ICON project involves producing databases of NDI techniques and procedures that will lead to the standardisation of inspection procedures (Dover & Rudlin, 1996; Rudlin & Dover, 1996). It is also foreseen that data-entry forms for inspection data in particular will become standardised. This will ease many problems in setting up inspection planning systems for practical use.

A usable IRM scheduling system would have to allow for these future requirements.

2.3 FATIGUE FRACTURE MECHANICS OF FIXED OFFSHORE STRUCTURES

The main mode of failure for fixed offshore structures in areas such as the North Sea is that of fatigue, that is, accumulative damage due to dynamic and cyclic loading. Fatigue damage manifests itself in the development of cracks or in the growth of existing small cracks in the structure. The main areas where cracks occur are at the welded tubular connections. Thus much of the inspection carried out is to detect and size cracks in the tubular joints of the structure. Similarly, much of the analysis carried out in evaluating the long term effects on a jacket structure of any damage is based on fatigue fracture mechanics.

This section outlines the analysis quantifying the damage caused by fatigue in structures. Much work on modelling the fatigue process in tubular joints has taken place in the NDE Centre, University College London, as part of large managed projects (Dover et al, 1986 and 1988) and as has been reported by Dharmavasan and Dover (1988).

2.3.1 The Process of Fatigue

The process of fatigue damage and its effect on ship and railway structures has been recognised since the mid-19th century. Since the 1940s, much work has been carried out to understand, model and quantify fatigue in general structures (Fuchs & Stephens, 1980; Pook, 1983).

Typical curves describing the progressive growth of a fatigue crack for different applied stress ranges are shown below, where S_1 is the highest stress range and S_3 is the lowest.

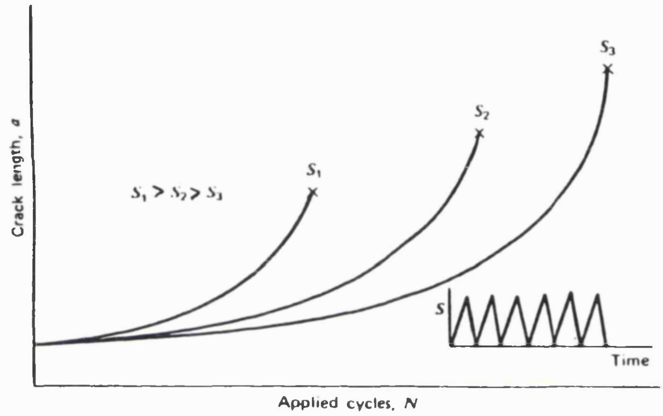


Figure 2.3 Typical fatigue crack growth (Fuchs & Stephens, 1980)

Broadly speaking, the life of a component being degraded due to fatigue can be divided approximately into three phases:

1. the crack initiation period, during which microstructural changes occur in the material leading to the microscopic cracks being formed, and that occupies most of the fatigue life of the component

2. the crack propagation phase, when macroscopic cracks grow at a stable and increasing but linear rate

3. very rapid or explosive crack growth leading to complete fracture of the component and which usually occupies a small percentage of the total fatigue life

Not all cracks grow to failure. A phenomenon known as *crack growth threshold* exists where, given certain conditions, cracks may reach a maximum size and grow no longer. It is not usually possible, under realistic conditions, to predict when crack growth threshold will occur. Usually then, it is assumed that fatigue cracks will grow to failure at some point in time.

The division between the phases listed above is somewhat arbitrary and, in fact, the idea of dividing the crack growth process into distinct phases only came about in the 1950s. It is useful in the

the crack growth process into distinct phases only came about in the 1950s. It is useful in the analysis of the crack growth as it allows simplified models to be employed, particularly for the 2nd phase, to predict crack growth with some certainty. In the case of welded structures, the existence of microscopic cracks is assumed and crack growth predictions are based on the 2nd phase only.

Fatigue assessment is carried out at various stages of the life of a structure: at design, in the interpretation of inspection results and in the process of justifying life extension. There are several approaches to fatigue assessment, that is, mathematical analysis, standardised or codified procedures, and service-load testing. In this section, it is the first approach that is considered in detail. It is important to realise that fatigue is a random process in that the exact size and shape of fatigue cracks cannot be predicted with accuracy. This is due to reasons such as material inhomogeneity, the uncertainties in the exact geometry of the component, varying loads and load paths and the model uncertainties inherent in the fatigue assessment.

2.3.1.1 Loading Modes

The way in which the crack will grow is dependent on several aspects. For instance, cracks will usually initiate at the points of highest stress, which implies that any discontinuities in the surface and any areas of corrosion are likely to contain the first signs of fatigue cracks. Small and neighbouring initial cracks will coalesce and will usually propagate in the plane of maximum tensile stress.

There are three modes of crack extension or, equivalently, of loading. Mode I is the opening mode where the load is tensile. In mode II, the load applied is a shear load in the direction of the crack depth extension, and in mode III, the load is a shear load in the direction of the crack length extension. Naturally, load combinations can and do occur.

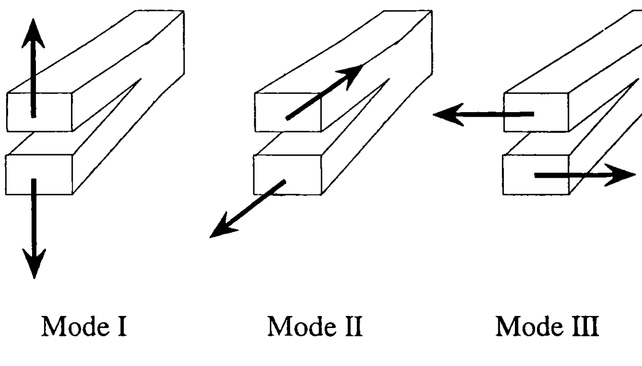


Figure 2.4 Mode direction and notation

The direction and plane of growth, although it is largely dictated by the initial crack and the mode of loading, cannot always be predicted exactly due to combinations of modes of loading.

2.3.2 Modelling Fatigue

Several established models are employed for evaluating the fatigue life and predicting future crack growth used in mathematical analysis of fatigue.

2.3.2.1 Miner's Rule

Early work in fatigue testing was carried out under constant amplitude loading. The problem of how to employ the results to predict the fatigue life of a component under variable amplitude service loading then arose. It was suggested by Palmgren in 1924 and Miner in 1945 (see Pook, 1983) that damage accumulates linearly with each cycle and at each stress level. This can be formulated by considering the damage due to one cycle at stress range S say, for which N is the number of cycles to failure, as

$$\Delta D = \frac{1}{N} \tag{2.2}$$

The total accumulated fatigue damage for a component, then, is given by:

$$D = \sum \frac{n_i}{N_i} \tag{2.3}$$

where n_i represents the number of cycles in the stress range S_i say, and where N_i is the number of cycles to failure for the same stress range S_i. This is the Miner-Palmgren model, also known as Miner's Sum or Miner's Rule. In design, this model of damage accumulation is the most common deterministic method used to predict the fatigue life of a component. The assumption made is that fatigue damage occurs with every stress cycle and that the accumulated damage is independent of the sequence in which the stress cycles occur. Using this approach, fatigue failure occurs when:

$$\sum \frac{n_i}{N_i} = 1 \tag{2.4}$$

In practice, a safety factor is applied so that the required sum value giving the maximum allowed accumulated damage in an evaluation, and thus the end of life, is less than unity. So as an example, if a safety factor of 2 is applied, then fatigue failure would be assumed to occur when the sum value equalled 1/2. Miner's rule may be used for simple loading cases to estimate the elapsed fatigue life of a component in situations when its past in-service load history is known. With a prediction of the future service load, it can also be used to estimate the remaining fatigue life.

2.3.2.2 Stress-Life Curves

A graph of the stress, usually the alternating stress S, as log(S), versus the life of a component, usually as log(N) where N is the number of cycles to failure, is commonly known as an *S-N curve.*

This curve indicates the life of component for a given load. Figure 2.5 shows a typical S-N curve for a smooth specimen under stress control test conditions, where failure is fracture of the specimen.

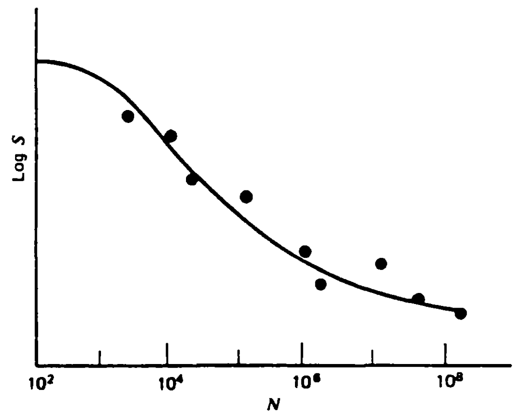


Figure 2.5 Typical S-N curve (from Pook, 1983)

The test results would show some scatter; typically the curve would be drawn through the median points to provide the smooth curve shown. A smooth S-N curve is typical of nearly all materials. One exception is that of low strength steels, particularly in corrosive environments, where discontinuities in the curve would be evident.

The S-N curve does not distinguish among the three phases of crack initiation, propagation and fracture, yet it does provide the total life to fracture at different levels of stress. This concept of an *effective S-N curve*, where the S-N curve from constant amplitude loading is normalised such that the Miner's sum gives 1, was introduced in the 1950s.

2.3.2.3 Loads and Stress Intensity Factors

For more advanced methods of predicting crack growth, it is necessary to be able to model the stress applied through the component and in particular near the crack tip, the point at which the crack is growing.

The stress intensity factor (SIF) is a convenient measure used to describe the stress field near the crack-tip. The exact form of the expression used to evaluate the SIF, or K, varies according to the geometry of the component and crack and the loading mode. For a 2-dimensional crack of size a (which may be either depth or ½ the length along the surface) in an infinite lamina subjected to a tensile stress S, K is given by:

$$K=S\sqrt{\pi a} \qquad (2.6)$$

For the more general case, a calibration factor Y may be employed in which case the SIF is written as (Kirkemo, 1988; Dharmavasan & Dover, 1988):

$$K = YS\sqrt{\pi a} \tag{2.7}$$

where S is the applied or nominal stress, and here, a is the crack depth at its deepest point, Y is a dimensionless factor and is dependent on the local geometry of the crack (such as the curvature of the notch tip) and on the loading mode.

Usually three different SIFs, K_I, K_{II}, K_{III}, for each of one the loading modes described earlier, may be calculated. In the general case, Y can be obtained by analytical methods for simple notches, but for semi-elliptical cracks, the evaluation of Y requires applying weight function techniques (some of these require the aspect ratio), finite element analysis or experiments or tests.

Once the SIF is found, the actual stress at any point close to the crack tip is now easily calculated. The stress is only dependent on the SIF, the distance r to the crack tip and the angle θ that the line from the crack tip to the point in question makes with the crack growth direction, as shown in the figure below.

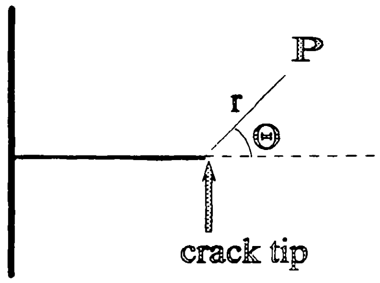


Figure 2.6 Use of SIF to calculate the actual stress at a point P

Several parametric solutions for the SIF for cracks in flat plates exist. Examples include the Newman-Raju model, which is based on finite element analysis results and assumes a semi-elliptical crack (Newman & Raju, 1981), and the Holdbrook-Dover solution for realistically cracked plates (Holdbrook & Dover, 1979).

The complex geometry of tubular joints does not allow analytical solutions for SIFs to be found and the flat-plate solutions give over-conservative results when applied to tubular joins due to the effects of load shedding. Thus two important practical SIF models for tubular joints are the AVS, or AVerage Stress (Dover & Dharmavasan, 1982; Dharmavasan, 1983), and the TPM, or Two Phase Method, (Kam, 1990) methods. Both are based on actual experimental results from cracked tubular joints.

2.3.2.4 The Erdogan-Paris Relation

This relation, also known as Paris' Law, describes crack growth rate for steels in terms of ΔK, the local stress range (Paris & Erdogan, 1963):

$$\frac{da}{dN}=C(\Delta K)^m \tag{2.9}$$

where a represents crack measurement, length or depth, and N is number of stress cycles and so represents time, ΔK is the stress range and C and m are constants.

This law is derived from typical log-log plots of da/dN against the local stress range such as is shown in Figure 2.7. It is a typical sigmoidal curve that can be divided into three regions corresponding to the phases of fatigue crack growth described earlier. The mid region showing a linear relationship between log(da/dN) and log (ΔK) gives Paris' Law.

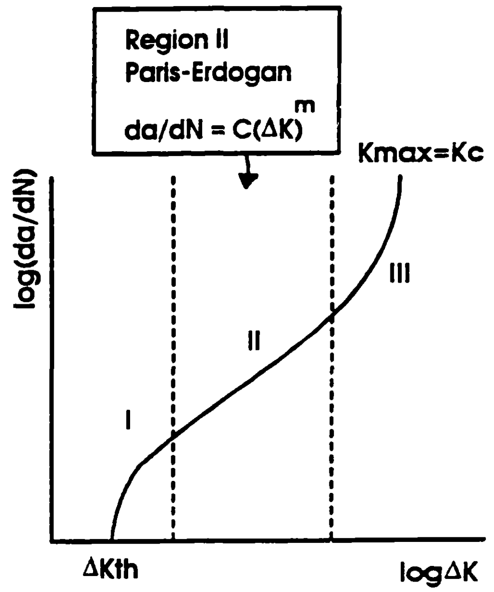


Figure 2.7 Typical fatigue crack growth versus local stress range (Dharmavasan et al, 1994)

The parameters C and m are dependent on the material. For steels used in fixed jacket platforms, m has a value of approximately 3 and C around 5 x 10^{-12}. The units of C depend on the actual value of m and given that the stress range ΔK has units MPa $m^{1/2}$ and da/dN is in m/cyc. The local stress range is calculated as follows:

$$\Delta K = K_{max} - K_{min} \tag{2.10}$$

As K is undefined in compression, K_{min} is taken as zero in the Mode I loading case. In other loading modes, this may not be the case. Fatigue crack growth is dominated by Mode I loads and hence other cases are not often considered.

2.3.3 Fatigue Life of Offshore Structures

Currently two main methods of predicting fatigue life of tubular joints in offshore structures are employed (Dover et al, 1991):

1. Use of stress-life curves with a damage assessment procedure

2. Fracture mechanics analysis

The first approach assumes that most of the life of the component is crack initiation. In contrast, fracture mechanics theory describes crack growth once a crack is present, that is, it does not predict when a crack will grow from a perfect specimen. The period of crack initiation cannot be taken into account using standard fracture mechanics theory. Ignoring crack initiation has the effect of making the procedure more conservative and, as such, may not be too undesirable. Even so, standard fracture mechanics theory can be used to predict the future degradation of a tubular joint, given an inspected crack size. Finally, fatigue fracture mechanics does not model the final fracture phase.

2.3.3.1 Hot-Spot Stress and Stress Concentration Factors

As fatigue cracks grow in areas of maximum stress, these areas are of particular interest. In tubular joints, the point on the intersection with maximum stress is termed the hot-spot. The stress at this point, the hot spot stress, is, in effect, the amplification of the nominal stress due to the geometry of component; it is a fictitious measure in that it cannot be physically measured. Thus to obtain the hot-spot stress, a stress concentration factor (SCF) is used. This is a factor by which nominal stress is multiplied to give the hot-spot stress value σ_{HS} or actual stress:

$$\sigma_{HS} = \sigma_n \, SCF_G \tag{2.11}$$

$$\text{Actual stress} = \sigma_{HS} \, SCF_N \tag{2.12}$$

where $SCF_G = f(\alpha,\beta,\gamma,\tau)$ is the geometric SCF, $SCF_N = f(1/r)$ is the notch SCF. The parameters are r, the radius of curvature of the crack at the notch tip, and α=2L/D, β=d/D, γ=D/2T, τ=t/T are non-dimensional, each representing a different aspect of the geometry of the component, since L=chord length, D=chord diameter, d=brace diameter, T=chord thickness and t=brace thickness. The MTD report on underwater inspection gives methods of interpolating for SCF values from experimental and tabulated values (MTD, 1989).

Use of Paris' Law for estimating the life of a tubular joint requires hot-spot stress values. To obtain these, first it is necessary to find an accurate SCF value. There are at least seven SCF parametric equations that are applicable to welded joints (Dover et al, 1991). The choice of which equation to use is not always obvious and the wrong choice can make a substantial difference to the estimate of remaining life, for example, up to +/-200%. To choose the most appropriate method, it is necessary to compare SCF values given by the equations with experimental values. Once the best fitting equation has been chosen, interpolation can then be carried out for unknown values.

A review of the SCF equations was carried out in the RISC project (Suganan, 1994). The Gibstein (GIB), Kuang et al (KUA), Wordsworth and Smedley (WS), modified Wordsworth and Smedley or UEG UR 33 (UEG), Efthymiou and Durkin (ED) and the Hellier, Connolly and Dover (HCD) SCF equations were considered. A statistical analysis was performed of the SCF predictions from the use of these equations and the results were compared to recorded experimental data on T and Y joints for

- both the chord and brace SCF irrespective of the validity range of the equation
- only the chord SCF with the validity range
- only the brace SCF also within the validity range

As can be seen in the Table 2.2, the results showed a wide variation for the first case which considered the full database of recorded SCFs for 173 joints. The variable being considered is the so-called bias factor that is the ratio of the recorded experimental SCF to the predicted SCF. Thus a bias factor of 1 is ideal, a value less than one implies over prediction and greater than 1 is under prediction. It would be expected that the results would be better for the cases within the validity ranges, but this was not always so. For example the Hellier, Connolly and Dover equation showed a mean value for the bias factor of 0.67, although with a low standard deviation of 0.11 for brace SCFs within the validity range, as compared to the mean value of 0.81 for all welded joints. This result may be due to the low number of recorded SCFs in the validity range, which for this example was only 9. Only a few atypical SCF values would have an undue influence the analysis results.

Table 2.2 Example SCF predictions for all welded members

	KUA	**ED(F)**	**ED(P)**	**WS**	**UEG**	**HCD**
Mean	0.95	1.13	1.01	0.86	0.83	0.81
S.D.	0.35	0.63	0.33	0.29	0.29	0.32

2.3.3.2 The Effect of Corrosion

Extensive experimental work at University College London, NDE Centre, has shown that for realistic modelling of the fatigue crack growth in a corrosive environment, a multi-segment crack growth rate versus stress range curve is required (Kam, 1990). Figure 2.8 shows a typical multi-segment curve.

Each straight segment of the curve represents a different set of material C and *m* values for the application of Paris' Law. Thus, estimating fatigue crack growth in a tubular joint in real environments requires these different sets of values together the values on ΔK axis or da/dN axis for which they are valid. This information does not exist for all required materials and are difficult to estimate.

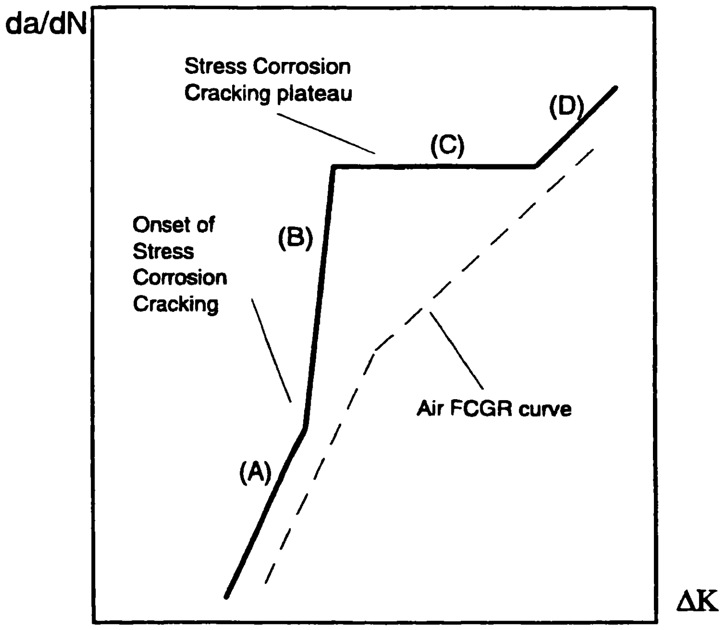


Figure 2.8 Multi-segment da/dN versus stress range ΔK (from MTD, 1989)

Work was carried out as part the RISC project to gather data on fatigue crack growth and then this was statistically analysed to find best fitting multi-segment curves (Bertini, 1994). A high-level materials database was created to contain the C and *m* sets for some materials in corrosive environments to be used by the RISC System.

2.3.4 Summary

To predict the fatigue crack growth in an offshore structure requires the use of a fatigue fracture mechanics crack growth model such as Paris' Law. Such models require data on the applied loads and the material properties. For realistic results, the implementation of the model must allow the effects of corrosion and the variation of the loading data to be taken into account. In addition, careful consideration should be taken of the model uncertainties inherent in the Paris' Law which ignores the early and late crack growth phases.

2.4 CURRENT IRM SCHEDULING AND PLANNING METHODOLOGIES

The scheduling of IRM actions is in essence a decision problem. It is necessary to carry out some form of quantitative analysis that will identify solutions to this decision problem, that is, to generate schedules, and give a measure of the effectiveness of each solution. In addition, the fatigue life of a welded connection, such as a tubular joint, is influenced by many uncertainties such as the wave loading, stress concentration factors, material properties and size and number of initial defects. The inspection and repair actions on the tubular joints are subjected to significant uncertainty due to the extremely difficult conditions under which they must be performed. It is important to incorporate or account for all these uncertainties into the inspection and maintenance plans for the structures.

IRM planning approaches in the offshore industry, such as the use of the Worst Case Growth Curve, attempt to take into account uncertain information and the difficulties of maintenance procedures. One method of incorporating uncertainties about the structure and the inspection process is to use reliability concepts as a decision tool for inspection planning. Some example IRM planning approaches from various industries are reviewed in this section.

2.4.1 Failure Mechanisms

The main potential modes of failure for tubular joints are yielding, buckling, punching, fracture, and fatigue damage. Each of these modes must be considered at each potential failure point on the structure. The combination of a failure point and an associated failure mode is termed a failure element, for example, one possible failure point is the hot spot of the weld of a tubular joint at which a fatigue crack may grow.

From a computational point of view, not all failure elements can be considered when estimating the reliability of an offshore structure. Those elements with the greatest probability are considered. Of all the failure modes, fatigue is considered one of the most important maintenance problems in the North Sea. Work on the reliability of offshore structures has been concentrated on the modelling of fatigue crack growth (Madsen et al, 1987a; Wirsching, 1984; Kirkemo, 1988).

2.4.2 Inspection Reliability

One major uncertainty is due to the inspection itself. Early work carried out within the aerospace industry led to the concept of a detectable crack size (a_{NDI}) for a non-destructive inspection (NDI) technique to be used with fracture mechanics and crack growth analyses (Rummel et al, 1988). The detectable crack size is defined as the minimum defect size that the NDI technique can detect with a high probability of success or detection. The concept corresponds to the basic requirement for an inspection technique, namely that cracks larger than a given critical size be detected virtually every time (Moyzis & Forney, 1982).

More generally, the reliability of an NDI technique has been traditionally quantified as the probability of crack detection (POD) which is dependent on the crack size (Berens & Hovey, 1982). This definition of reliability allows the more usual case that there is no clear detectable crack size to be modelled. The POD of an inspection technique is affected by crack qualities other than length or depth, for example position, or by human factors, and the required confidence level.

Another general way of describing the ability of a technique to detect is to consider randomness of the detectable crack size. Thus the reliability of detection may be modelled as a distribution of detectable crack size. Confusingly, the same term, POD, is sometimes employed by reliability experts for this description. Regardless, in the experimental work carried out on the reliability of NDI techniques and in manufacturers' specifications, POD refers to a probability and not the distribution of detectable crack sizes.

Usually after detection, further work, such as re-inspection and sizing, may be carried out to confirm and provide information for crack assessment. Reliability of sizing or probability of sizing (POS) will be needed for a complete assessment, and the POS cannot be derived from the POD data alone (Kam & Dover, 1987).

The false call rate, that is, the probability of a nonexisting crack being detected, sometimes termed the probability of false indications (PFI), should also be taken into account. The accept/reject decision level affects the detection capability of the technique; if all indications are accepted then the POD and PFI values for the technique will be high. Conversely, if few indications are accepted, so eliminating all spurious indications, the PFI and the POD will be low. A high PFI value is, of course, undesirable since unnecessary repairs would be carried out, while a high POD value is very desirable. Therefore a careful balance has to be made in choosing the acceptance decision level for reasonable values of both the PFI and POD (Wall & Wedgewood, 1994).

In the ideal situation, the three factors, represented by POD, POS and PFI values, need to be considered to make a thorough assessment of inspection techniques.

2.4.3 Load and Stress Histories

The loads experienced by offshore structures are inherently random, since they are due to winds and waves and thus require modelling as such. Since the loads applied are random, the responses produced in the structure are also random. Many models exist that describe random load histories.

These range from intricate techniques modelling exactly the loads over a period, to methods attempting to condense the loads and responses to one load which produce damage equivalent to that caused by the actual loads. This topic is a complex one and is outside the scope of this thesis. It suffices here to explain very briefly the two basic approaches to the modelling of wave forces (McClelland & Reiffel, 1986) and describe how they are used in the modelling of the responses.

In the deterministic approach, the wave forces are calculated through the detailed consideration of a sequence of maximum wave heights representing the variability of the sea. The sequence itself can be treated statistically. A detailed analysis of loads due the wave height still requires many simplifying assumptions. The alternative approach involves the representation of the sea as power spectra, incorporating the variability of the sea directly into the calculations. It involves making the assumption of linear superposition, which does not hold true in all situations.

With power spectra, it is possible to model *sea-states* that are periods during which there is a constant characteristic wave. By this it is meant that the distribution representing the loads due to waves is invariant with respect to time, that is, a sea-state can be described as a stationary stochastic process. A sea-state is defined by the *significant wave height* that is a characteristic value for the wave height, and the wave dominant period, that is a characteristic value for the period of the wave. In addition, the power spectra may take into account the directional nature of the waves. To complete the model of a changing sea, a scatter diagram is used to represent the joint occurrence of the significant wave height and dominant time period. Power spectra representing different states of load history can be obtained from dynamic stress analyses and can be used directly in fatigue analysis. These calculations can be time-consuming, since they involve generating the stress history by simulating the waves and hence the stresses experienced by the structure, then counting the resulting stress cycles and assessing the fatigue damage.

More rapid methods of carrying out fatigue assessments under random stress histories exist (Kam & Dover, 1988). These involve calculating an *equivalent stress range* that is, an imaginary constant amplitude stress range causing the same total fatigue damage over the same number of cycles as the random stress history. The equivalent stress range method requires some modification if it is to be used for prediction. The random nature of sea-states requires the equivalent stress range to be described statistically by use of the stress range probability density function (SRPD).

2.4.4 Other Factors

2.4.4.1 Repair Actions

The effects of repairs carried out *in situ* may not be as expected. For instance, since underwater welding is very difficult, it is possible that the crack being repaired is replaced by worse defects in the new weld. It is also possible, although not likely, that grinding carried out to remove a detected crack may, by reducing the thickness of the member, reduce the strength of the node. Certainly, the reduced thickness does lead to a shorter period before new cracks grow to through-thickness.

For these reasons, the choice of when and how to repair, requires careful consideration. The changes to the node caused by a repair require modelling in some form.

2.4.4.2 Costs and Measures of Utility

In the problem of rational IRM planning, the cost, or other utility measure, for each action and the consequence of failure are required. Unfortunately, predicting these exactly is not always possible. There is no systematic method of considering how to incorporate uncertainties in costs into the required analyses, save perhaps by the use of *fuzzy logic*, which is described in Chapter 4. Ad hoc methods may be employed, such as not attaching great importance to the exact values and considering a range of costs to be approximately equivalent. For instance, a simplistic example is that of comparing two plans with associated values of £150000 and £150500. A difference of £500 could be considered negligible and thus the two plans would have equal worth. This simplistic approach does require deciding on the allowable ranges, that is, the *granularity* has to be defined. Fuzzy logic may overcome some of these problems, but as explained in Chapter 4 it introduces other difficulties.

2.4.4.3 Gross Errors and Human Factors

Although reliability methods provide a more rational basis for inspection planning, accounting for gross errors within the analysis is difficult. For most work in this area, it is considered that these errors are accounted for within the procedure to carry out the inspection and to report the results. The procedures are set up to attempt to minimise the effects of any mistakes by divers and top-side operators of inspection equipment.

2.4.5 Current Work in Inspection Planning

Several different approaches have been investigated which seek to account for the uncertainties described above. The approaches vary from deterministic analyses, such as the Worst Case Crack Growth which uses upper or lower bounds for the variables of interest, to pure stochastic methods which make full use of distributions of random variables and time variance of loads. In addition, they differ in terms of the variables taken into account. For instance, the Receiver Operating Curve procedure, also known as the Relative Operating Curve, considers in detail the effect of the false call rate and the POD for an NDI technique, but does not use other factors explicitly. On the other hand, general structural reliability analysis methods attempt to consider all uncertainties. Other procedures emphasise a few of the factors, while trying to include other possibly less important factors. Some possible approaches to aid inspection planning are described here.

2.4.5.1 Worst Case Crack Growth

One methodology, which is essentially deterministic, uses a worst case crack growth curve and the idea of a detectable crack size (a_{NDI}) for an NDI technique. A worst case crack development curve is found by carrying out analyses using subjective choices of bounding values for the input parameters, that is, for the stresses, and the material properties.

By not allowing cracks to grow beyond the critical size, a_{cr}, within one inspection interval, a maximum allowable starting size or a maximum interval between inspections can be determined. The starting value determines the minimum allowable a_{NDI} and hence the NDI techniques which may be used for a particular inspection interval. An example curve is shown in Figure 2.9.

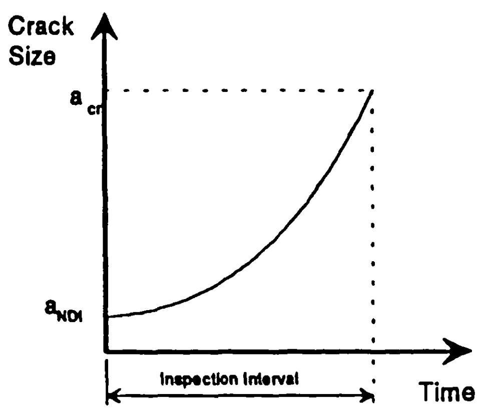


Figure 2.9 Crack growth and the a_{NDI} concept

This approach has been extended to provide flexible definitions of a_{NDI} and to allow variable inspection intervals (see Figure 2.10), but it is restricted to considering only a few of the total uncertainties (Kam, 1988).

A disadvantage of this type of semi-deterministic approach is that the predicted crack growth could be made excessively conservative since only bounding values are used. The lack of a common measure of fitness makes it difficult to compare the reliability of different components. Another point to note is that these approaches are detection-based strategies. Fracture mechanics is only used for determining a_{NDI} and not for in-service assessment of defects. In-service crack growth assessment is required in the offshore industry, since repair is not always an efficient solution in the short term, since the size of the offshore components and the environment mean that repairs may be of low quality.

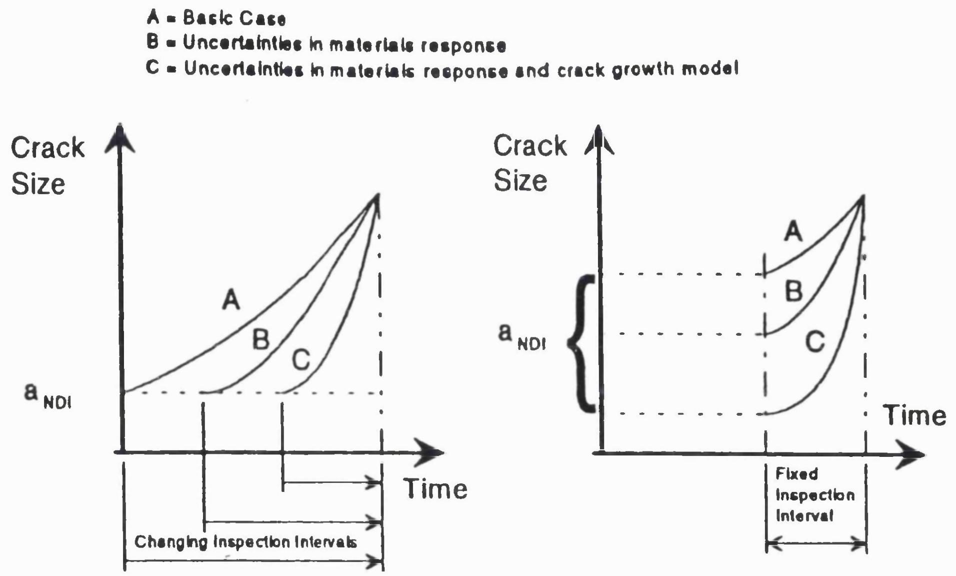


Figure 2.10 Inspection planning using the a_{NDI} approach

2.4.5.2 The Relative Operating Characteristic (ROC) Method

This decision tool takes into account variation due to human operator skill and makes an economic argument based on the capabilities of inspection techniques only (Spanner, 1988; Kam, 1989a). In medicine, a similar technique is known as Receiver Operating Curve (Wall & Wedgwood, 1994).

As the criteria for a successful detection varies from exacting to less demanding, the POD for an inspection technique increases. The false call rate or PFI, however, will also increase, although usually at a slower rate. A Relative Operating Characteristic (ROC) graph can be obtained by plotting the POD versus the PFI for an NDI technique as shown in Figure 2.11. A measure of merit for the technique is given by twice the area between the curve and the diagonal, that is, the integral of the POD with respect to the PFI. For the perfect technique, that is, one which will detect all cracks, POD = 1 irrespective of the detection criteria, and so the area is 0.5. For a useless technique, the PFI is the same as giving correct information, that is, PFI =POD and the area is 0.

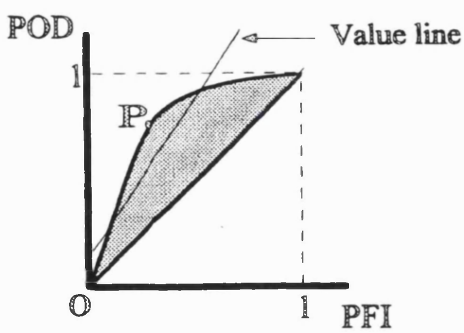


Figure 2.11 A typical Receiver Operating Curve with Value Line

The economic decision required is to ensure that employing the technique adds value to the system. The value of the inspection can be calculated by considering the number of spurious indications and of both the missed and detected defects, obtained from the original population of defects, with associated costs. A straight Value Line, representing no-added-value, can be plotted on the ROC graph. Thus the points on the ROC which lie above the Value Line give positive added value, and the point furthest from the line, point P in the diagram above, represents the maximum added value.

Indirectly, the ROC method may take fatigue, or other failure modes, into account by using fatigue fracture mechanics, or other failure models, to decide what is a potentially damaging defect and to evaluate the consequences of missing a defect. Furthermore, crack growth predictions may be used to plot several Value Lines on the ROC graphs, where each line represents a new point in time and therefore a different assumed population of cracks. The point with the best overall added value can be identified from the family of curves. This point lies on the Value Line corresponding to the optimum inspection time and would indicate the optimum accept/reject criteria to be adopted.

2.4.5.3 Fatigue Reliability

Another method, proposed by Connolly (1994), uses fatigue fracture mechanics and its starting point is the distribution of initial crack or defect sizes present in the structure after fabrication. The aim of this method is to use POD values directly as defined by the NDE specialists, as opposed to the distribution of the detectable crack size. The method is applied to one component at a time and takes no account of future inspection or repair actions.

Other very recent work considering fatigue is reported by Zimmerman and Banon from Exxon (1994). The method applies simple system reliability concepts, as will be described in Section 3.2.4, to the structure and probability of fatigue failure is computed using the Monte Carlo technique.

2.4.5.4 The MSG -3 Method

This is an example taken from another industry. MSG is an analytical methodology developed by the Air Transport Association of America for the implementation of a reliability centred maintenance strategy (Brascamp, 1991). The third version, MSG-3, was introduced in 1980.

MSG-3 is employed to define, at the design stage, an initial maintenance strategy for a structure. If a satisfactory maintenance strategy cannot be defined, then redesign of the structure is required. During the life of the structure, the data and information arising from maintenance tasks are fed back into the maintenance strategy.

The approach is structured and systematic: the first step is to identify maintenance elements:

components or subsystems of the structure that can be treated as an individual item on which some maintenance task will be required. The next step is to divide the strategy into two areas of concern.

One area considers the failure effects or consequences for each maintenance element. The procedure at this stage is similar to the failure mode effects analysis (FMEA) approach of classical reliability theory. Failure modes are classified according to whether or not they can be easily detected within the normal operation of the structure and according to the type of consequence, of which there are two in MSG-3: safety failures leading to a hazardous situation and operational failures leading to an economic problem.

The second area of concern is that of identifying maintenance tasks and their order of preference. For instance, carrying out routine maintenance before repair is preferable. In certain situations and for certain components, some tasks may not be feasible. For example, inspection may be the only routine task that may be carried out on certain irreparable components, and at some point replacement may be the only suitable option.

MSG-3 determines practicable maintenance strategies for a structure and produces valuable maintenance plans at the design stage. Such a pragmatic philosophy allows what is a fundamentally difficult problem to be tackled and at least partly solved.

2.4.5.5 Risk and the ALARP Principle

The Health and Safety Executive in the UK has applied a systems approach to general safety management in many industries (Birkinshaw et al, 1993). This approach considers the organisation required of the corporate body to achieve the safety objectives set; the identification and assessment of the hazards and risks; the application of safety plans; the monitoring and control activities; and, finally, the audits and reviews of the safety procedures and the strategies employed. Notwithstanding this, setting appropriate safety targets requires an understanding of levels of risk.

The accepted definition of risk is given as

$$\text{Risk} = \text{POF x Consequence} \tag{2.13}$$

where POF is the probability of a failure event and the Consequence is usually expressed quantitatively as the total costs associated with the failure and provides the context. The failure costs can include the cost related to closing down the structure after the event, the often very much lower repair cost, and, at the other extreme the cost associated with injuries and fatalities. Although the last are hard to quantify, figures have been published by HSE and others that have been used in cost-benefit analyses of public projects and these vary according to the type of project.

For any decision problem involving a choice between courses of actions, there is clearly some point at which the risks are perceived to be unacceptably high and the action is rejected; conversely, for very low risks, the decision is to carry out the action. In the mid-region, the decision becomes complicated. The HSE uses the term As Low As Reasonably Practicable (ALARP) to indicate this mid-region where reducing risks to a minimum practicable level is desirable. Setting of the upper and lower bounds for the ALARP region is dictated by regulatory and social considerations.

The ALARP principle makes explicit the idea that there is a grey area where risk may be tolerated, if it has been considered and reduced wherever possible. Reduction of the levels of risk is carried out by balancing the costs of carrying out a safety action against the marginal reduction in the costs associated with the risk.

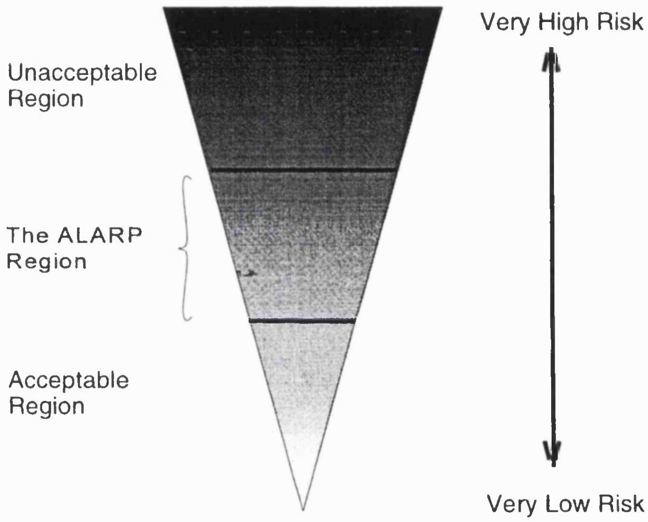


Figure 2.12 The ALARP principle

2.4.5.6 Structural Reliability Methods

In the last decade, modern reliability methods have become accepted tools in several industries such as offshore, nuclear and aerospace. The potential of modern reliability methods as decision support tools in engineering applications lies in the consistent modelling of uncertainties. These methods provide the means to compare the consequences of different actions and designs, as in the fatigue design of offshore structures (Dalane et al, 1990).

In more recent years, the application of modern reliability methods within the framework of classical decision theory to inspection and maintenance planning of engineering systems subject to uncertain deterioration processes has been investigated (see Thoft-Christensen & Baker, 1982; Faber et al, 1992). The work has led to a consistent framework for evaluation of the consequences, as expected costs, of carrying out different inspection and repair actions. The evaluation of expected costs allows optimisation of the overall inspection and maintenance plan for a given structure within its anticipated lifetime.

Other examples of the full use of structural reliability is in the work by Det norske Veritas (Lotsberg & Marley, 1992) and by WS Atkins (Gierlinski et al, 1993). DnV has now introduced inspection planning based on target reliability measures. These are based on annual probability of failures and crack growth as the only measure of fatigue damage. Inspection can result in no detection or detection followed by measurement of a crack. Updating can be carried out with direct measurements of stress and gross errors are not accounted for directly. WS Atkins carry out full reliability assessments of fixed jacket platforms using the software RASOS.

The reliability of the structure could be maintained in several ways, such as by carrying out repair operations at various scales or more frequent inspections. The costs and benefits of these options could be assessed in part by making use of reliability-based fracture mechanics analysis.

2.5 DISCUSSION AND CONCLUSIONS

In this chapter, current inspection procedures have been reviewed. The information from the Working Group and the survey of the available documentation of inspection procedures provided the basis for the requirements specification of a decision support system to scheduling of IRM actions. The acquisition of information and knowledge took place over many meetings and it was clear early on that the information gathering process would be iterative. The results confirmed the problem described by Christer et al (1989) that it was difficult to identify the exact decision-making procedure in IRM of platforms. Several versions of a report describing the inspection procedure, in here Sections 2.1 and 2.2, were circulated among the members of the Working Group until it was agreed that the information it contained was representative of what actually happens in the operator organisations.

The background to fatigue fracture mechanics has also been described in this chapter. Fatigue fracture mechanics theory provides the fundamental concepts required to understand the mechanism behind the degradation of a jacket-type platform due to dynamic wind and wave loads, and then to predict the crack growth in the tubular welded connections of a platform. This becomes the main mechanism for predicting the state of the structure for the purpose of planning future IRM actions.

A short review of current approaches to maintenance planning has also been given. Included in this was a very brief outline to the concept of reliability-based planning and some factors to be taken into account. Structural reliability analysis is considered to be a promising method of providing decision-makers with objective measures by which to base their IRM planning policies.

The three elements considered in this chapter were combined to provide the base assumptions for a Reliability based Inspection SCheduling (RISC) methodology. The procedure was restricted to consider only the sub-deck and the planning of inspections of the tubular welded connections of steel jacket-type platforms. Only fatigue crack growth is considered as the damage mechanism, since fatigue of tubular joints is considered the main area of concern to both the operators and the certification authorities.

To implement the RISC methodology, a computer-based knowledge base system, the RISC System, was proposed which not only carries out numerical analysis but will aid planning of IRM actions taking into account the constraints on operators.

This chapter has provided the background to the problem of IRM planning. A quantitative framework is required to make rational and objective decisions based on the assumptions here given. In the final section, it was hinted that structural reliability analysis is a suitable framework for IRM planning, hence a detailed description of structural reliability and decision theory as applied to scheduling of IRM actions, follows in Chapter 3.

3 RELIABILITY ANALYSIS OF OFFSHORE STRUCTURES

Solving the decision-making problem of when, where and how to carry out inspection actions has traditionally involved using engineering judgement. A more objective and rational way of evaluating structures is to use structural reliability analysis that takes into account the uncertainties inherent in the structures and the environment. There is a need to incorporate reliability into a system view of offshore structures for the evaluation of inspection, repair and maintenance strategies. A system approach requires the use of several techniques and, to enable their application, an automated system that can integrate all the various forms of information, the methods to be used, the structural model and data, and on appropriate IRM strategies. Such a system could present candidate IRM plans to the operators of the structure.

This chapter reviews methods for evaluating the reliability of structures. The fundamental case is described and the techniques used to evaluate the reliability of more complex cases are explained. The use of structural reliability analysis in the offshore industry and the influence by reliability concepts to guidelines and codes are discussed.

The background to decision theory is outlined. The application of decision theory to the problem of IRM planning for fixed offshore platforms incorporating fatigue fracture mechanics for tubular joints is described. Decision theory provided a framework for an IRM analysis procedure that integrated reliability analysis with fatigue fracture mechanics analysis. The procedure was implemented as a set of software programs. The work on the analytical procedures was carried out in parallel to the design work for the RISC System described in Chapter 5.

3.1 WHY RELIABILITY FOR OFFSHORE STRUCTURES?

Although structural analyses, and so structural codes, have traditionally been based on deterministic analyses, there are now concerted moves towards reliability-based analysis that can take into account the inherent uncertainties in

- environmental conditions leading to loading variation, both from wave and wind action and from unknown conditions
- material properties, since steels cannot be wholly homogeneous
- structural geometry, as there are slight variations in diameters of joints etc.
- inspection results, since inspection techniques will not detect all defects and will not be completely accurate in sizing

- analysis models, which are approximations of the real world

Engineering makes use of quantitative models to analyse and evaluate structures. Some methods and techniques have become complex and highly elaborate, requiring input data of greater detail than may be available to be able to test the predictions made.

In an effort to account for lack of information or doubtful data, deterministic analyses in structural engineering usually employ safety factors, sometimes known as factors of ignorance (Gordon, 1978). An immediate consequence is that the structures so evaluated are by definition over designed. A more serious consequence is that the results gained from such an analysis do not provide any long term knowledge about the structure. In the past, failures in structures analysed in this way and known to have been constructed as-designed have been attributed to poor materials. Such reasoning does not hold for steel structures. Material properties of metals vary by no more than a few percent of the characteristic or mean values. Yet the safety factors employed in the design of steel structures are usually greater than 2 and hence are more than sufficient to counteract any variations in the material properties.

Historically, it was believed that single safety factors were sufficient. It was believed that practical absolute limits of load and strength existed and so it was only necessary to use a safety factor to account for uncertainties in construction, materials, etc. All that was needed to be able to use a safety factor only slightly larger than 1 was more knowledge on the behaviour of the structure, better models and more data.

Currently, it is now understood that for large complex structures in difficult environments, even complete knowledge can never lead to complete certainty (see Blockley, 1992). Probabilistic analysis is now part of an engineer's basic set of modelling tools and many textbooks exist on the subject, such as that by Ang and Tang (1975) and Jardine (1973). It is also generally accepted to be the best type of analysis (Brebbia & Walker, 1979; Madsen et al, 1987b). This is because it will give

- a measure of the uncertainty in the outcome of the evaluation of the structure
- indications of any major inadequacies in the modelling of the structure or of the data entered

The uncertainty measure is not important in itself, although it can be used to compare different designs or actions and to weigh the costs of adopting a course of action as part of risk analysis. More important in the long term is the second set of measures which allow targeting of problems to be tackled in the future. As data and models are updated, re-analysis adds to knowledge of the expected behaviour of the structure.

3.1.1 Types of Uncertainties

In dealing with uncertainties, Thoft-Christensen and Baker (1982) differentiate between four classes:

- Physical uncertainties, that is to say uncertainties in environmental conditions, hydrodynamic loading, material properties cannot be ignored nor easily reduced. For instance, in fatigue crack growth studies, it is now known the scatter of data is due to the inhomogeneity of materials (Kirkemo, 1988).
- Statistical uncertainties are possibly due to human errors, though mainly due to inaccuracies in measurement, or inadequate sampling (Spanner, 1986).
- Model uncertainties arise from attempting to quantify a problem. Modelling a system without making any simplifying assumptions is rarely possible. A model is an idealized representation of the system and thus making highly accurate predictions, no matter how accurate the input values are, is not usually possible. Also, unknown effects or unknown boundary conditions play their part.
- Gross errors can occur in design, construction, planning, analysis, etc., or arise from malfunctioning measurement equipment. Gross errors are usually the result of one-off events and hence are not easily modelled.

The first two of these types of uncertainties can be incorporated into probabilistic reliability analyses relatively easily and how this may be done is discussed below. Model uncertainties can also be considered if any approximations made can be identified. Gross errors, in contrast, are dealt with by considering procedures that will highlight possible problematic situations and so will not be covered in detail here (see Anderson & Kragh, 1991; Lotsberg & Marley 1992; Spanner, 1986).

3.2 GENERAL BACKGROUND TO RELIABILITY ANALYSIS

Reliability is the science of evaluating an item's fitness-for-purpose within its required life-span. A useful definition for reliability is given by British Standard BS4778 as

> "The ability (probability) of an item to perform a required function under stated conditions for a stated period of time".

There is also a generally accepted mathematical definition for reliability, although even this definition may appear in various forms,

$$R(\ g(\underline{x},\underline{a})\ )=1-P(\ g(\underline{x},\underline{a})<0\ ) \tag{3.1}$$

where R(.) is a measure of reliability as a probability and thus has a value in the interval [0,1], $\underline{x}$ is the vector containing all the relevant controlling parameters, usually including time, $\underline{a}$ is a set of limiting values to distinguish the states of the system, $g(.)$ is the state function of the generalised system under investigation defined such that if $g(.) < 0$ then the system is said to be in a state of failure, and P(.) is the probability of the event occurring, hence $P(g(\underline{x}, \underline{a}) <0)$ is the probability of system failure, sometimes denoted as POF.

The main purpose behind the study of reliability analysis is to attempt to identify common elements in the methods of dealing with uncertainties. Yet the definitions given above are too general and too abstract for practical use. There are practical difficulties in finding a quantifiable measure of reliability, which is appropriate for an application, and of deciding upon, and collecting the data for, the relevant parameters. Because of these difficulties, various interpretations of reliability have evolved in different industries, each developed according to the particular requirements for the relevant application.

The following sections explain the particular methods developed for structural reliability analysis with reference to offshore structures. Inspection and maintenance uncertainties are important to structural reliability when considering the lifetime of the structure. Fatigue fracture mechanics theory, which is specially relevant to fixed steel jacket-type platforms in the North Sea, can be incorporated into reliability analysis by defining failure in terms of the size of fatigue cracks in welded tubular joints.

It has long been recognised that the reliability of a structure is best judged from a systems perspective taking into account the interactions between the components making up the structure. Ideally, aspects such as the economics of construction, maintenance and long term management should also be included in the equation as made clear in very early commentary by Forsell from 1924 (Lind, 1970) and an early paper by Gordon (1957). These considerations can alter the way in which the methods are implemented and applied.

3.2.1 Uncertainties in Structures

Consider the strength (plastic moment, buckling strength, fracture toughness, etc.) and loading (maximum bending moment, buckling load, stress intensity factor, etc.) values in any structure or component (Figure 3.1). Both strength R and the loading S could be, and in the real world usually are, random variables. The failure of the component does not depend on the absolute values of the two variables, but on the difference between the two. That is, failure occurs when the loading is greater than the strength, whatever the values of strength and loads.

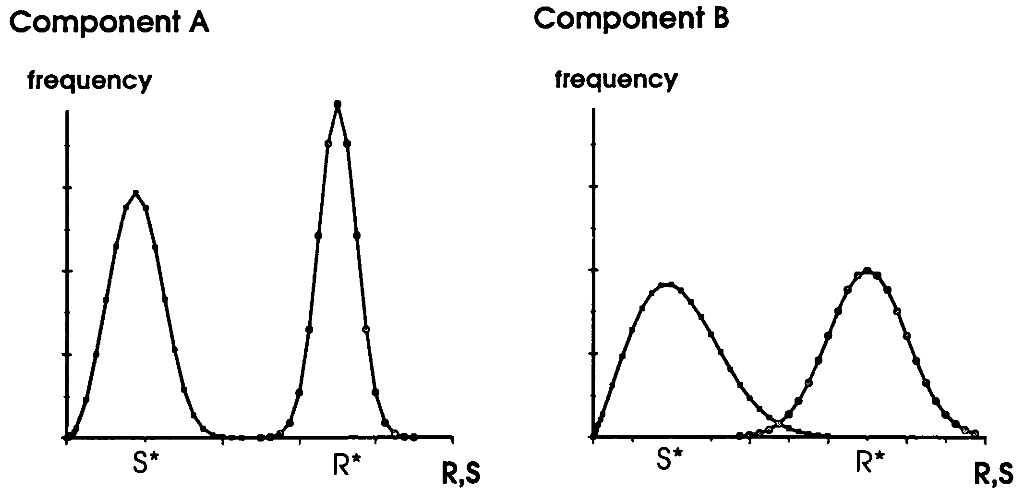


Figure 3.1 Loading and strength variations for two components

A simple assessment based on comparing some form of characterisation values, such as for instance the mean values, in Figure 3.1 represented by S* and R*, is not a complete assessment of the reliability of these components. For example, the difference between R* and S* for both the components is the same, but it is obvious that component A is more reliable than component B. The area of overlap suggests where the structure is unsafe. To quantify the reliability of a component, a more advanced approach for reliability assessment than just comparing the means is required.

3.2.2 The Limit State Function

The *limit state function* or *failure function* is introduced to formulate a universal approach for reliability analysis. The failure function $g(.)$ is defined to describe the failure of a structural component such that whenever $g(.)$ falls below zero, the component or system is then said to be in a state of failure, similarly whenever $g(.)$ is above zero then the system is said to be safe or functioning. For example, in a simple strength against loading situation, a *safety margin* M may be taken to be

$$M = R - S \tag{3.2a}$$

The corresponding limit state function is then

$$g(r, s) = r - s \tag{3.2b}$$

where r and s are realisations of R, the resistance or strength of the component, and S, the stresses or loads applied to the component.

The boundary $g(r, s) = 0$ is called the *failure surface* or the *limit state*. The probability of failure (POF) is given by

$$\text{POF} = P(\, g < 0\,) = \int_{-\infty}^{0} f_g(x)dx \;=\; F_g(\,0\,) \tag{3.3}$$

where $f_g(x)$ is the probability density function of $g(.)$ and $F_g(x)$ is the cumulative distribution function of $g(.)$.

More generally, the safety margin may be defined in terms of many basic variables $\underline{X}$, say, which in general are random, with a corresponding failure function $g(\underline{x})$, which divides the space of the basic variables into two regions: the *failure space*, given by $g(\underline{x})<0$, and the region of safety or where the component can be said to be functioning, given by $g(\underline{x})>0$. If the joint probability density function $f_{\underline{X}}(\underline{x})$ is known, then the POF can be calculated by noting that the POF is exactly the probability that the joint random variables are in the failure space:

$$\text{POF} = \int_{g(\underline{x})<0} f_{\underline{X}}(\underline{x})d\underline{x} \tag{3.4}$$

3.2.2.1 The Fundamental Case

Consider in more detail the case above when g has only two independent variables, R and S, but where it is known that each variable is normally distributed as shown in Figure 3.2. Given the well-known properties of linear functions of normally distributed variables, the resulting value of the failure function $g = r\text{-}s$ is also normally distributed.

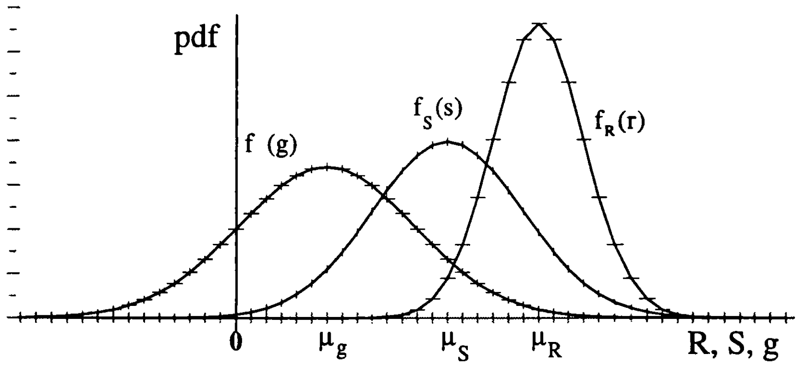


Figure 3.2 Normally distributed strength and loading random variables R and S

The mean and standard deviation of g are easily calculated by

$$\mu_g = \mu_R - \mu_S \tag{3.5a}$$

$$\sigma_g^2 = \sigma_R^2 + \sigma_S^2 \tag{3.5b}$$

Since failure occurs when $g < 0$, then the area under the graph of $f_g(g)$ for $g < 0$ in Figure 3.2 represents the probability of failure. From equation 3.4, and by standardising the normally distributed random variable g, the POF is given by:

$$POF = F_g(0) = \Phi\left(\frac{0-\mu_g}{\sigma_g}\right) = \Phi\left(-\frac{\mu_g}{\sigma_g}\right) = 1 - \Phi\left(\frac{\mu_g}{\sigma_g}\right) \tag{3.6}$$

where $\Phi()$ is the standard normal cumulative distribution function, which is tabulated.

If R and S are not normally distributed, then there are other combinations of distributions, such as lognormal and normal, lognormal and exponential, etc., where the exact POF can be calculated by using equation 3.3 in a similar manner. Furthermore, equation 3.4 may be used easily in cases where the joint distribution function is a product of the individual distribution functions, that is where the random variables are independent, and where the failure space can be clearly defined in terms of the limits of the multi-variable integral.

For the general case involving many non-normally distributed random and possibly correlated variables, the evaluation of equations 3.3 or 3.4 is usually very complex as it involves multi-dimensional volume integrals. Such integrals are often very difficult, if not impossible, to evaluate analytically. Because of this, alternative methods are required to estimate the probability of failure of a component.

3.2.2.2 <u>A Definition of the Reliability Index</u>

In order to overcome the problem explained above, the results of the Central Limit Theorem may be called into play. This theorem states that for non-normally distributed random variables X_1, X_2, ... X_N, the random variable Y, a linear combination of X_1, X_2, ... X_N, (that is, $Y = \sum a_i X_i$), will become normally distributed as N tends to infinity. The mean and standard deviation of Y can then be approximated by the values obtained from the corresponding expressions assuming normal distributions for X_1, X_2, ... X_N. The theorem in effect allows the approximation of the POF by calculating the ratio of the sample mean of g to the sample standard deviation of g, and then exploiting the relationship:

$$POF \approx 1 - \Phi\left(\frac{\mu_g}{\sigma_g}\right) \tag{3.7}$$

It has been found that this expression does provide a good estimate for POF in many practical situations where the failure function can be given as a linear combination of several basic variables. It also has the advantage of requiring the use of only the first two moments of the basic variables. Most of the available data or information on realistic structural performance is sufficient to evaluate only the first and the second moments for the population from the sample, that is, the sample mean values and variance of the respective random variables including the covariances of pairs of variables. Practical measures of safety or reliability, therefore, are necessarily limited to evaluations based on these first two moments.

As the ratio of the mean of g to the standard deviation of g is often easy to calculate given the limitations, and as the POF is easily estimated from this ratio, Cornell defined it as the *reliability index* (β_C) (see Thoft-Christensen & Baker, 1982):

$$\beta_C = \mu_g / \sigma_g \tag{3.8}$$

Another point to note here is that the assumption still being made is that the failure function is a linear one. In the case when the failure function is not linear, then the definition above has to be extended, but the idea behind Cornell's reliability index is the basis of what is known as a Level II technique. The various levels of structural reliability analysis techniques are next explained, before considering the general non-linear failure function.

3.2.3 Levels of Structural Reliability Analysis

Most structural reliability analysis is complex and many methods exist for idealising and simplifying structures for different types of problems. It is convenient to classify analysis methods into levels of approximation (Thoft-Christensen & Baker, 1982):

- **Level I**

These methods, which take only a characteristic value for each parameter, are termed Level I methods. They usually only provide a pass or fail result. Although not complete methods of reliability assessment, Level I methods can be used for very rapid assessment. Design codes that include a checking procedure are in effect Level I methods.

- **Level II**

Methods that employ only two values of each uncertain parameter, usually the mean and variance with some measure of correlation such as the covariance, are termed Level II methods. These methods employ approximations and iterative procedures to calculate an approximation to the probability of failure.

- **Level III**

Techniques used to carry out full reliability analysis, without any idealisations taking place as to the random variables involved, in particular in terms of joint distributions and the failure region, are classified as Level III methods. The resulting POF values are accurate, given model uncertainties and errors in data collection. As indicated in the above simple example, full Level III analysis is usually very difficult unless the state function is composed of normally distributed random parameters. Therefore, in practice and in general, Level I or II methods are used. Level III methods may be employed for the purposes of the calibration of values used in the approximate methods.

- **Level IV**

For completeness only, it is fit to mention this classification which is sometimes mentioned in literature (Gnedenko & Ushakov, 1995). Level IV methods are in essence Level III methods incorporating economic models. The application of such methods is also known as risk analysis.

3.2.3.1 Overview of Level II Analysis

In Level II methods, the exact POF is not evaluated directly. Instead the method seeks to find a measure of reliability that can give an estimate for the POF in some way. In Section 3.2.2.2, the use of Cornell's reliability index was explained for estimating the POF for a system with a linear failure function. In this section, the case of nonlinear failure functions is now considered. To understand the idea, the simplest and fundamental case of S and R, two normally distributed variables, and the failure function $g = r\text{-}s$ is again considered:

A set of transformed variables is calculated:

$$Z_1=\frac{R-\mu_R}{\sigma_R} \qquad Z_2=\frac{S-\mu_S}{\sigma_S} \tag{3.9}$$

The state function is then transformed into

$$g=r-s=\sigma_R z_1-\sigma_S z_2+\mu_R-\mu_S \tag{3.10}$$

For the limit state ($g = 0$), equation 3.10 describes straight lines both in the space of the basic variables R and S and in the transformed space of Z_1 and Z_2 (see Figure 3.3).

From analytical geometry, the shortest distance of this line from the origin is

$$|OP|=\frac{\mu_R-\mu_S}{\sqrt{\sigma_R^2+\sigma_S^2}} \tag{3.11}$$

This is exactly the reliability index as defined by Cornell for this case of a linear failure function and normally distributed random variables. When the failure function is not linear then Cornell's reliability index, although useful and practical, has no physical basis.

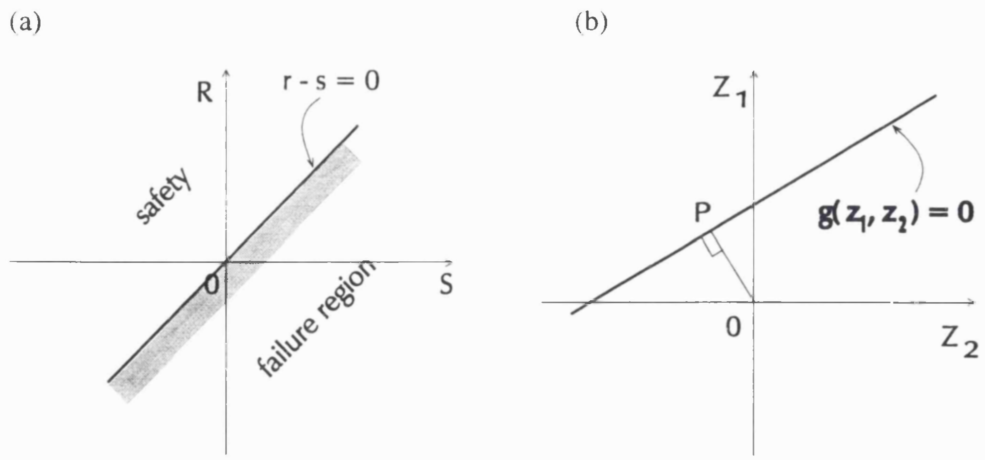


Figure 3.3 Linear failure surfaces (a) in the R, S space and (b) in the Z_1, Z_2 space

The development given above does lead to a more stringent definition by Hasofer and Lind of the reliability index (β) or the *safety index* as the shortest distance from the failure function to the origin in the space of standardised normal random variables. This is given by

$$\beta = \min_{\underline{z} \in g(\underline{z})=0} \sqrt{\sum_i z_i^2} \tag{3.12}$$

where the z_i represent the standard normal basic random variables on the failure surface, that is, on the line $g(\underline{z}) = 0$. The point P on the failure surface which is closest to the origin is termed the *design point*. The distance from the design point to the origin is a more convenient parameter to describe the reliability of the strength versus load system and other more complex systems, as it remains invariant for the same failure space.

To explain the relevance of invariance of the reliability measure, consider the same fundamental problem as before: two normally distributed random variables R, representing strength, and S, representing loads applied. The safety margin M= R - S has already been discussed. The form of the safety margin is clearly a choice: another reasonable choice for the safety margin, with its corresponding failure function, is

$$M = \ln(R) - \ln(S) = \ln(R/S) \tag{3.13a}$$

$$g(r,s) = ln(r/s) \tag{3.13b}$$

This safety margin is as appropriate as M=R-S, since its corresponding failure function $g(r,s)$ would satisfy the conditions that $g(.)<0$ for failure and $g(.) > 0$ for safety.

The Cornell reliability index for this failure function can be calculated to be:

$$\beta_C = \mu_g / \sigma_g = \frac{\ln\left(\frac{\mu_S^2}{\mu_R^2}\sqrt{\frac{\mu_R^2+\sigma_R^2}{\mu_S^2+\sigma_S^2}}\right)}{\sqrt{\ln\left(\frac{(\mu_R^2+\sigma_R^2)(\mu_S^2+\sigma_S^2)}{\mu_R^2\mu_S^2}\right)}} \tag{3.14}$$

and this is clearly different from the expression for Cornell's reliability index obtained for the safety margin M = R - S.

If the space of the basic variables R and S is considered, the limit state or failure surface for the case M=R-S will be the same for both safety margins since the same set of points will form the failure region, as shown in Figure 3.3(a). The reliability index as defined by Hasofer and Lind will give then the same result for different but equivalent failure functions. The property of invariance of the equivalent failure functions is very desirable and so this definition is more often preferred.

The reliability index approach is the basis of the advanced Level II formulation that uses only the first and second moments of the random variables to estimate the probability of failure for a structure (Thoft-Christensen & Baker, 1982; Thoft-Christensen & Murotsu, 1984).

The exact relation of the reliability index to the probability of failure, for the case of normally distributed basic random variables and a linear failure function, is exploited to estimate the POF for all other cases:

$$POF \approx 1 - \Phi(\beta) = \Phi(-\beta) \tag{3.15}$$

where $\Phi(.)$ is the standard normal cumulative distribution function.

In general then, if the state function consists of linear, uncorrelated and normally distributed basic variables, Level II and Level III analyses would give the same result.

3.2.3.2 The General Level II Method

The approach discussed above has been extended to study the interference of multiple (more than two), correlated, mixed normal and non-normal random variables and non-linear state functions. As the argument is a geometric one, the technique is based on finding this minimum distance in the failure space of the standardised, normalised random variables, which will require approximating the failure surface by a tangent surface at each point. Thus, Level II analysis is iterative.

The steps taken for Level II analysis are:

1. Transform the given system of basic variables into a set of normalised and standardised

random variables

$$\underline{X} :\to \underline{Z} \iff \underline{Z} = T(\underline{X}) \tag{3.16}$$

2. Approximate the failure surface given by $g(\underline{Z}) = 0$ by a tangent surface $h_r(\underline{Z})$ that is more easily analysed, such that at a point P_r, which has coordinates $\underline{Z}_r$ close to the design point P, $h_r(\underline{Z}_r) = g(\underline{Z}_r)$

3. The surface $h_r(\underline{Z})$ is then used to find a new point P_{r+1} closer to P, which in turn is used to define a new $h_{r+1}(\underline{Z})$.

4. The process repeats from step 2.

3.2.3.3 First and Second Order Reliability Method

If the tangent surface $h(\underline{Z}) = 0$ is a hyper plane, that is, if the failure surface is linearised, the method is called First Order Reliability Method (FORM) (see Figure 3.4). If $h(\underline{Z}) = 0$ describes a hyper parabolic surface then this is Second Order Reliability Method (SORM).

SORM generally leads to quicker convergence to the design point. It has the disadvantage of being computationally more expensive at each step.

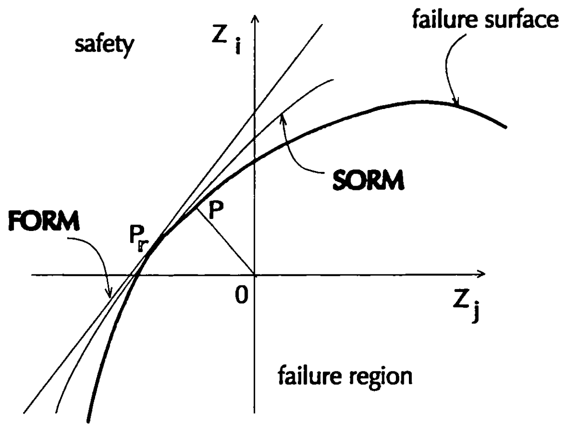


Figure 3.4 Tangent surfaces to the failure surface at point P_r

The approximation to the failure surface is easily achieved by considering a Taylor's series expansion of the failure surface and taking the constant and terms in z_i in the case of FORM, or terms up to and including z_i^2 for SORM.

The definition of the reliability index by Cornell is often in practice a good approximation to the more formal Hasofer and Lind definition. Hence, calculating the ratio of the mean to the standard deviation of the limit state function is often carried out for a first estimate of the reliability of a component. Of course, if the state function is normally distributed, then Cornell's reliability index provides an exact measure of reliability.

3.2.4 Extending Structural Reliability Analysis

The SORM and FORM techniques employed for evaluating the reliability index for a structure are approximations. In spite of this, the basic procedure can be computationally intensive if employed on complex cases. Because of this, avoiding full Level II analysis is desirable whenever possible.

Two basic ways of extending the reliability analysis techniques discussed can be considered:

- complex failure modes for components
- systems of components

The following explains methods that help to reduce the computational effort.

3.2.4.1 Updating the Reliability Index

One method of reducing the need to carry out component level analysis, is to take advantage of *Bayesian updating* of the reliability index. Updating can be employed when an evaluation of the reliability index has already been carried out, but new information on the structure requires a re-evaluation of this index. One example is that of using a recent inspection result for a tubular joint to update the reliability index for after the inspection.

The basic idea behind Bayesian updating is the application of Bayes' Theorem, which is given by:

$$P(A_r|A)=\frac{P(A|A_r)P(A_r)}{\sum_i P(A|A_i)P(A_i)} \tag{3.17}$$

where A is a subset of the world of possible exclusive events $\{A_i\}$. The event A could represent the result of an inspection of a component and the {Ai} could be the state of the component, where for instance, A_1 = failure and A_2 = safety. Here the state events are exclusive, since the component cannot fail and be safe simultaneously. An inspection result is a subset of {failure, safety}, since failure or safety and inspection results are not mutually exclusive and, of course, {failure, safety} covers all possibilities.

For Bayesian updating of the probabilities to be applicable, the following conditions must hold (Turkstra, 1969):

1 All possible future events can be listed.

2 There exist well-defined relative probabilities for all events.

3 There exist measures of the undesirability of all events.

In other words, for every action A_i, all possible outcomes or events E_{ij} with their probabilities $p(E_{ij})$ and costs u_{ij}, can be given. The total cost for A_i is given by

$$T_i = \sum p(E_{ij})u_{ij} \tag{3.18}$$

From this, the total cost, T_i, is an expected value.

In the situation where the absolute probabilities of failure or safety, $P(A_1)$ and $P(A_2)$, are known and the probability of any inspection result, given the failure or safety of the component is known, $P(A|A_1)$ and $P(A|A_2)$, then obtaining the probability of failure given an inspection result $P(A_1|A)$ is simple. Thus, the updating procedure allows the consideration of conditional probabilities, that is, given new information, the reliability index can be updated without having to carry out the lengthy methods described earlier.

3.2.4.2 Systems Reliability

The above discussion has concentrated on component level reliability analysis. For structures, taking a systems point of view, that requires being able to evaluate the reliability of the system as a whole is important. Systems reliability evaluation requires a consideration of the effect of failure of each component and of every combination of components on the reliability of the structure.

Systems modelling can be carried out in various ways. One of the most commonly used approaches to evaluating the reliability of structural systems is to consider *block diagrams* (Bentley, 1996). There are two fundamental ways of combining components according to reliability considerations:

- Series structure, where all components in the system are required to carry out the task, that is, failure of any one component leads to system failure.
- Parallel structure, where components share the task and hence all components have to fail for system failure to occur.

Block diagrams are drawn to represent the combinations as shown in Figure 3.5.

These diagrams do not necessarily represent how the components are physically joined. For example, a parallel connection of capacitors would represent a series system in a reliability sense, as each capacitor needs to function for the system to function. Another example is that of the legs of an offshore structure. Physically, these are approximately in parallel, but if failure is defined as an event requiring cessation of production to allow major repairs to take place then the loss of any one leg would constitute failure. Thus, the main legs of a jacket platform represent a series system.

(a) (b)

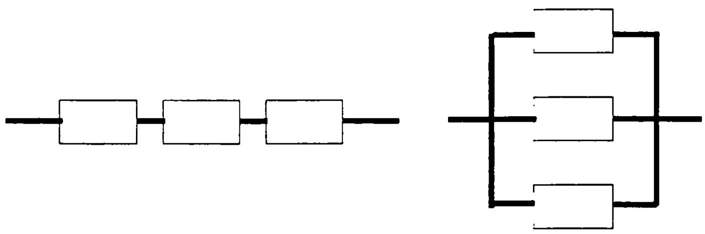


Figure 3.5 Components combined (a) in series and (b) in parallel

At this point it is important to note that, as before for reliability analysis at component level, the definition of failure needs to be stated clearly. Returning to the example of the legs of a platform, if failure were redefined as the formation of a mechanical hinge in the frame, i.e. total collapse of the structure and as failure of one leg would not cause a hinge mechanism to form, then the legs of a six-legged platform would form a parallel system.

Parallel systems are often employed to provide *redundancy*. A fixed offshore platform is a redundant system since groups of components share the task of supporting a load. The effect of redundancy is to increase overall reliability. Fail-safe design is based on these considerations, since failure of a component will not lead to the immediate failure of the whole system. Series systems, on the other hand, are the most common in practical engineering terms, in that it is usually required that all components of a system function. Failure of one joint in a platform may not lead to collapse of the structure, but in terms of evaluating maintenance, the need for repair may be considered to be failure and if repair must always be carried out, the platform is in effect a series system.

The reliability of a simple parallel system can be calculated relatively easily given the reliability or POF of each component and that the failure of each component has no effect on the failure of the other components. The probability of failure POF for a parallel system is represented by P(F_1 and F_2 and .... F_n) where F_i is the event that the i^{th} component fails. If events are independent then,

$$POF=P(F_1 \text{ and } F_2 \text{ and } \ldots F_n)=P(F_1)\text{x} \ldots P(F_n)= \prod_{i=1}^{n} P(F_i) \qquad (3.19)$$

The reliability of a simple parallel system is then given by

$$R=1 - \prod_{i=1}^{n} P(F_i) \qquad (3.20)$$

The probability of failure for a series system is represented by P(F_1 or F_2 or ... F_n) where F_i is the event that the i^{th} component fails. The assumption of exclusivity is not a sensible one as failure of any one component does not preclude failure of any other. Therefore, this probability of failure is not easily derived directly. Instead it is easier to consider the reliability of the system R =

P(system functions or is safe). Since a series system functions if all its components function or are safe, R is given by P(S_1 and S_2 and ... S_n) where S_i is the event that the i^{th} component is safe. So for a simple series system,

$$R=P(S_1)\times ...P(S_n)= \prod_{i=1}^{n} P(S_i)= \prod_{i=1}^{n} \{1-P(F_i)\} \tag{3.21}$$

Pure series or parallel systems rarely occur in practice. In fact, for parallel systems, often more than one component needs to function for the system to function. These are termed *k-out-of-n* structures: where *k* components must function for the system with a total of *n* components to function. A simple example of this that of a system of 3 pipelines carrying a fluid, each capable of carrying 100 gallons per second. Thus, if all are functioning, a total of 300 gallons per second can be carried. If it is expected that a minimum of 150 gallons per second must be carried, then at least 2 pipelines must be functioning. This is then a 2-out-3 structure. The evaluation of such a structure is more complex, but usually not impossible.

To model complex systems, the two simple series and parallel models are combined. It is relatively easy to calculate the reliability of a pure *series-parallel systems*, that is, a series system of parallel subsystems, or a *parallel-series systems*, a parallel system of series subsystems.

(a) b)

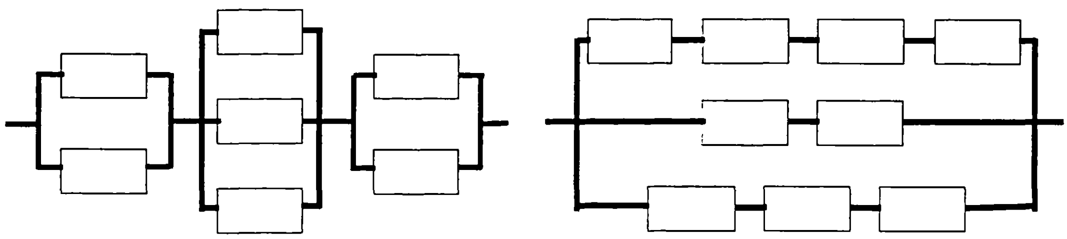


Figure 3.6 (a) Series - parallel and (b) parallel - series systems

An example of a combined system is that of two computers monitoring a process. The computers form a parallel system since if one fails the other one can still carry out the task. Each computer could be considered a series system of components, since if any one of a computer's components fails then the whole computer fails. For jacket platforms, which are redundant frame structures, it is usual to model these as a series system of highly critical substructures. The substructures are systems of components carrying out a load-sharing task.

In the idealised situation, where all failures are independent and where the parallel systems are purely parallel, that is, only one component needs to function for the parallel system to function, then the calculation of the reliability for a given series-parallel model of a structure is straightforward. In the real world and non-ideal situation, three issues arise:

- Failures of components, such as joints, are rarely independent, since failure of one joint usually indicates that the environmental conditions differ from those assumed during design and this would have a bearing on the rest of the structure.

- There a very few pure parallel systems; out of a group of components sharing a load, it is usually the case that more than one component is required to carry the total load.

- Defining a suitable model of the structure is not an easy task, as it requires identifying all the sets of components which share loads, that is all possible combinations of failure leading to failure of the structure.

To add to the above complications, a block representing the failure of one component may appear in two or more parallel subsystems, in which case, the assumption of independence is no longer reasonable.

Given a model, it is possible to calculate upper and lower bounds for the POF by making use of assumptions of independence and pure parallel systems. These methods are beyond the scope of this thesis, but may be found in any textbook considering systems reliability (for example, Ang & Tang, 1984) and also in the literature considering structural reliability (Thoft-Christensen & Murotsu, 1984).

Additionally, an approximation to the model is often acceptable. Instead of attempting to identify all parallel subsystems, only the most important are identified and sometimes a maximum number of components per subsystem are considered. The simplification reduces the computation involved in calculating the total POF for the structure. Unfortunately, the problem remains of identifying the components and parallel subsystems making up the approximate model for structure. Considered components and subsystems are those whose failure has a substantial impact of the POF of the structure and have high POFs, and thus they are critical to the system. Any method that identifies the important components will also carry out some evaluation of the POF for the structure and require some information on the POF of the component.

Methods of estimating the POF for a structural system and identifying the series-parallel model include two techniques that are specific to structural reliability analysis:

- **β-Unzipping**

In this method developed at the University of Aalborg, each component's reliability index is calculated using the FORM and SORM techniques (Thoft-Christensen & Baker, 1982; Thoft-Christensen & Murotsu, 1984).

The technique exploits the concept of approximating the systems model as a series-parallel system with a maximum number of components per parallel system. If the maximum number is 1 then this results in a pure series system; if 2, then the model is a series system of parallel systems with only two components; and so on. This is termed *systems modelling at level N*, where N is the maximum number of components allowed. Thoft-Christensen and Baker (1982) use level 2 modelling for fixed platforms, since the POF of a 2-element parallel system can be calculated easily even when allowing for dependencies.

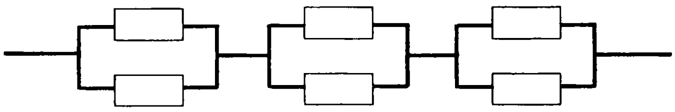


Figure 3.7 Level 2 modelling of a structure

The method constructs a systems model at level N as an approximation to the actual structure and at the same time calculates the reliability of the model. The model is created by identifying the components which have the greatest impact on the reliability of the structure. The procedure first considers the reliability index values for all the components. Certainly the components with lowest β values are to be considered for inclusion. A maximum β value is required to limit the number of components considered. This value is set in an ad hoc fashion by considering the total number of components and using experience to decide on a reasonable proportion to be included for a particular structure. For each of these components, its failure is simulated and the new reliability of all remaining components is calculated. The β values for each combination of two failures can now be calculated and any with a β value higher than the maximum is discarded. If the process stops at the point of identifying only important pairs of components, the result is a level 2 systems model. Assuming exact POFs for each component, the exact POF for this model can be calculated. This is not the exact for the real structure, but assuming good approximations for the POF of each component, the level 2 model POF often gives a reasonable approximation to the POF for the actual system. Continuing the process allows the structure to be modelled to any required level N.

The β-Unzipping method can be extended by considering ways in which a sensible choice of the definition of the range of the reliability index values can be made. One interesting example of extending this approach is given by Chen et al (1996).

- **Branch-and-Bound**

This technique is taken from operational research and has been applied to the same problem at the University of Osaka Prefecture (Thoft-Christensen & Murotsu, 1984). It is a more rigorous approach and more complex to carry out. The idea behind this technique is to find the most likely failure paths, that is, the sequence of components which fail such that eventually the structure fails. A failure path is modelled as a parallel system of elements, representing components with associated damage mechanisms, which are linked to form a series-parallel model for the structure. Failure paths which are likely to have negligibly low POFs are discarded before attempting to calculate their POF.

If a failure path has more than 2 elements, then an upper bound is calculated for the POF of the failure path. This process of considering combinations, involves calculation of the upper POF for each combination, selection of the most likely, and *branching* again from this combination. Once a most likely complete failure path has been identified, using the upper value for the POF as an estimate, the lower bound for the POF of the complete parallel system is calculated and is used as a *bounding* value. Since the POF for a parallel system is always lower than the component POFs, it is not necessary to be concerned about subsystems guaranteed to have a very low POF. Thus the components or combinations of components with an upper bound for the POF that is lower than the bounding value can be discarded and need not be considered any further. By considering each incomplete failure path from the set already identified while finding the first failure path, the process is repeated until the next complete failure path is found. A lower POF is calculated for this and if this is larger than the current bounding value, it is taken to be the next bounding value and used to discard failure paths with very low POFs. By this process, the most likely failure paths, that is, the parallel subsystems with highest POFs, can be identified with an upper and lower value for their POFs. From this, it is simple matter to calculate a lower and upper value for the POF of the series system made up of the failure paths and so of the structure.

Several variations on branch-and-bound exist, such as the Selective Enumeration Method that orders components according to their importance to the structure, before considering each in turn for inclusion into the failure path (Shetty, 1994). These variations try to avoid carrying out labourious calculations required in enumerating the possible failure paths, which are later discarded, by reducing the number of possible branches in the first place.

- **A Pragmatic Approach**

As even these techniques are computationally intensive, it is more usual to take a pragmatic approach that takes into account the consequence of failure. For instance, the failure of a highly critical node, such as a connection at a leg of an offshore structure, will be of greater concern than the failure of a small X-joint at the centre of the structure. Hence a lower reliability for the smaller node will be more acceptable than for the node at the leg.

Some formal techniques that systematically identify the consequences of failure exist, such as failure trees and Failure Mode and Effect Analysis (FMEA) (Bentley, 1996). For jacket platforms, this type of analysis is already carried out by certification authorities for defining the criticality of individual tubular nodes (MTD, 1989). Consideration of the consequence of the failure dictates that different minimum threshold values of the reliability index be set for different types of nodes. For critical nodes, higher levels of reliability are required than for non-critical nodes. The scheme takes into account the redundancy of the structure and therefore in some form the system reliability.

3.2.5 Applications of Reliability Analysis in the Offshore Industry

Reliability analysis has been applied to many industries to provide a rational assessment of the structures based on uncertainties introduced by manufacturing processes, testing procedures, material properties, the design models employed and also the in-service conditions. Some examples of the application of reliability concepts can be found in

- manufacturing and production industries, in which reliability analysis is employed to decide planned maintenance strategies and customer service provisions (Bentley, 1996)
- the aerospace industry, where probabilistic analysis is carried out during design (see an example of a reliability structural optimisation package in Rajagoplan & Grandhi, 1996) and reliability centred maintenance is now routine for army aircraft in the US (Anderson & Neri, 1990)
- construction, where optimal design considers reliability of the completed structure (Rojiani & Bailey, 1984)
- marine industries, where reliability centred maintenance is also now routinely applied and interest has been shown in design evaluation based on reliability (Faulkner & Sadden, 1978; Baker & Vrouwenvelder, 1992)

The various forms of probabilistic analyses are carried out in many different industries for applications in three areas, namely, evaluation at the design stage, planning of maintenance, and reevaluation after an incident.

Marine structures suffer the added problem of dynamic loading. The huge uncertainties inherent in the environment, in addition to the uncertainties common to any other structure, need to be considered in analysis. Hence, the application of reliability analysis has great interest to the offshore industry. Currently, detailed reliability analysis at component level is carried out for re-evaluation of structures after incidents. Reliability analysis considering the ultimate load-carrying capacity of a structure is used in the re-assessment of structures for life extension or increased top-side loading (Stewart et al, 1988). Systems reliability is also considered in the design and operation of offshore structures (Baker & Vrouwenvelder, 1992).

3.2.5.1 Design Codes

Classification authorities have been incorporating structural reliability methods into their codes of practice and guidelines since the late 80s. The Load and Resistance Factor Design (RP 2A-LRFD) code produced by the American Petroleum Institute makes use of structural reliability methods to calibrate the factors used (API, 1993). LRFD is in effect a Level I method (Winkworth et al, 1993).

The classification authority Det norske Veritas Classification takes a different approach (DnV, 1982). It allows and recommends the direct use of Level II methods, although it recommends the use of Level II methods, in particular the Monte Carlo technique, to calibrate the results of the Level II method applied. Since different classification authorities have taken such different approaches, some recent moves have been made towards producing an international standard for the application of structural reliability which attempts to incorporate all these approaches (Wirsching, 1985; Snell, 1993).

3.2.5.2 Evaluation of Designs

Structural reliability analysis is being applied as part of detailed re-evaluations of tubular joints and other components after incidents and as part of detailed design. Much work has been carried out in applying probabilistic concepts to fatigue design of tubular joints (Wirsching, 1984; Wolfram, 1986).

One interesting example of probabilistic assessment of oil platforms, which extends quality control concepts, is given by Saldanha Peres and Rogerson (1984). This work made use of a distribution of fatigue crack sizes and numbers of cracks in a platform to derive a probability of failure for a population of similar platforms. This is rather different to most other applications of reliability concepts for platforms, where assumptions of similarity are not exploited and indeed avoided. Such examples can be found including reliability analysis for re-qualification and design of jacket platforms (Diamantidis, 1986; Diamantidis et al, 1991; Dalane et al,1990; Frieze, 1989) and of tension leg platforms, considering fatigue and including inspection updating (Ximenes & Mansour, 1991).

3.2.5.3 Probabilistic Analysis For Planning

Interest in the application of reliability and risk analysis to the maintenance problem has increased over the past ten years. The reasons for this include the realisation that many structures in areas such as the North Sea have been over-designed. For instance, recent fatigue research in tubular structures has shown that large fatigue cracks can be tolerated sometimes in joints within the design life (Dover et al, 1986 and 1988). In addition, underwater operations for large scale maintenance, such as detailed inspections or repairs in the form of re-welding, replacement or grouting, are very costly. Thus, maintenance schedules produced with the aid of deterministic analyses may lead to unnecessary and expensive repairs. Recent research results have provided the type of data required to be able to apply reliability analysis to offshore structures.

In Norway, maintenance planning for offshore structures by DnV is carried out based on target reliability measures for the year (Lotsberg & Marley, 1992). It is considered that crack growth is the only measure of fatigue damage. Updating, however, can be carried out with direct measurements of stress and inspection is assumed to result in either no detection or detection followed by measurement of a crack. Although reliability methods provide a more rational basis for inspection planning, gross errors are not accounted for within the analysis and thus need to be accounted for in the planning procedure.

One approach uses the idea of *fitness-for-purpose*: if the components in question can be shown to be fit for their intended purpose, then no maintenance action is required. Use of reliability analysis to establish a relative level of *fitness* (such as described in Kam, 1988, and Kirkemo, 1988) allows targeting of the more critical joints to be considered for repair. Further, rational planning of future remedial actions can also be carried out based on measures of expected level of fitness. If repair is preferred for all joints in which a defect has been found, the analysis should also show whether the operations could be delayed until a time of better weather. A possible added benefit of working under less hostile weather conditions is that of higher quality inspections and repairs. Another possibility is to reschedule the inspection programme to maintain the required level of reliability and thus eliminate the necessity for repair, at least in the short term.

The option to delay could give industry more flexibility in rationalising and targeting expensive maintenance operations. The next sections discuss this application in more detail and consider the use of reliability analysis in RISC.

3.3 Structural Reliability Analysis for IRM Planning

From the review of structural reliability in Section 3.1, it follows that if IRM planning is to be based on considerations of reliability, advanced Level II reliability techniques may be employed. In addition, models of fatigue crack growth should be included in the analysis, as was discussed in Chapter 2. This section discusses the integration of fatigue fracture mechanics with structural reliability, explains the decision problem behind inspection and maintenance planning, and describes the structural reliability analysis programs implemented for RISC.

3.3.1 A Limit State Function for IRM Planning

For offshore structures with tubular members, it is common practice to define accumulated damage failure as the occurrence of a *through-thickness crack*, that is, a crack that has grown through the thickness of the tubular member. An appropriate safety margin is of the form:

$$M = \text{Member Thickness} - \text{Crack depth} \tag{3.22}$$

It is convenient to generalise (3.22) to consider a critical crack size, instead of member thickness, and a general crack size at a time t. The limit state function for this case becomes simply

$$g(x) = a_{CRIT} - a(t,x) \tag{3.23}$$

where a_{CRIT} is the critical crack size and $a(t,x)$ is the actual crack size at time t. This limit state function is often reformulated as

$$g(x) = N(a_{CRIT}, x) - N \tag{3.24}$$

where N is the number of elapsed stress cycles and $N(a_{CRIT}, x)$ represents the number of cycles to failure.

To calculate the crack size at any point in time requires fatigue fracture mechanics and in particular Paris' Law as explained in Chapter 2. Analytical integration of the crack growth relation from an initial defect a_i to the final size a_{CRIT} gives the lifetime in terms of the number of cycles to failure for the joint. In theory, a similar integration of the crack growth relation over a number of stress cycles, or equivalently (assuming an average number of cycles per unit time) over a period, will give the crack size. In practice, the integration is not possible, since the SIF, ΔK, in general is related to the crack size. Hence, a numerical integration procedure is required. The numerical integration procedure can be applied to any crack growth curve and so to a multi-segment curve, which allows incorporation of corrosion information.

Fatigue crack growth models for welded structures, based on linear elastic fracture mechanics, have been described recently by several authors (Dijkstra & Straalen, 1991; Dharmavasan et al, 1991; Thorpe, 1986). The IRM planning of offshore fixed platforms requires detailed consideration of tubular structures and therefore the general fatigue fracture mechanics model is modified for steel welded joints.

The real geometry of the tubular joint is translated to a simplified model as given in Figure 3.8. Fatigue cracks in tubular joints generally start at the weld toe and grow both in depth and width into the tube wall, and are assumed to be semi-elliptical as shown in Figure 3.9.

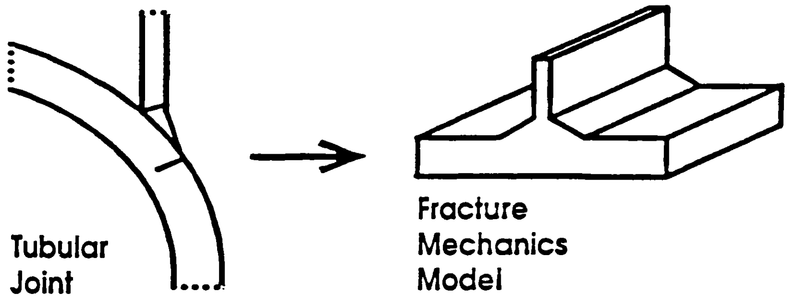


Figure 3.8 Fracture mechanics model for tubular joints

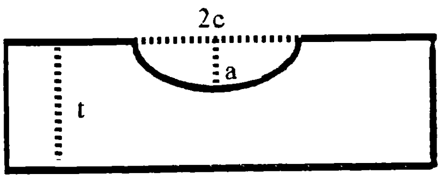


Figure 3.9 Semi-elliptical surface crack shape

A model that describes the 2-dimensional shape of the crack is required, since such a model can be used with inspection results given as crack lengths only. Some have been proposed, such as that suggested by Ma and Kam (1991). In the RISC project, two methods that carry out a transformation of length to the depth based on crack extension rules for both directions were considered:

1. Using crack growth models for both crack depth and width. Paris' Law can be used for crack growth in both directions, independently, with the appropriate SIF range for each direction.

2. Applying a *forcing function* for crack width direction. Paris' Law is used only in the depth direction and the crack width is given as a function of the crack depth.

The first approach is more accurate from a theoretical point of view. In practice it is more difficult to implement as there is no obvious way of incorporating the effects of multi-initiation and crack coalescence, since in practice several short cracks will join to form one long crack (Dover et al, 1994).

The forcing function for the second approach relates the depth to the length. The available forcing functions were derived from experimental observations, but it is not yet certain that they are applicable to situations different from the experiments on which they were based (Dijkstra et al, 1994). In the short term, the use of crack length as an indicator is not viable and it is best to consider, when possible, only crack depth.

3.3.2 Optimal Maintenance Planning

The goal of the rational IRM planning methodology is to provide a tool for the identification of an inspection and maintenance plan which can fulfil the practical requirements. An optimal inspection and maintenance plan is one that yields the minimum expected total costs for maintaining the structure throughout its anticipated lifetime. As the number of practical constraints imposed on the inspection and maintenance scheduling is very large, carrying out automatic identification of the optimum plan is not always possible. This is because the total number of numerical operations necessary to estimate the expected costs for all inspection and maintenance plans can be extremely large and time consuming.

One method of overcoming these problems is to employ an adaptive scheme for inspection and maintenance planning (Fujita et al, 1989). In this scheme only the next inspection time t_{insp} is planned based on a selected inspection method and repair strategy. When the inspection has been performed, the next inspection is planned taking the most recent observations into account.

3.3.2.1 Overview of the Use of Decision Theory

IRM planning of offshore structures can be formulated as a decision theory problem. In practical decision problems, such as inspection and maintenance planning for offshore structures, the number of alternative actions can be extremely large and a systematic analysis of the corresponding consequences is necessary. The following is based on the theory in Sorensen et al (1991).

The decision problem can be stated as: which experiment e should be chosen from the space of possible experiments E? and, given the result from experiment e, what action a should be performed out of the possible available actions A? It is known is that experiment e yields a random result r of possible experiment results R, and action a will in turn have a consequent effect on the state of the universe, resulting in random state s, one of a possible set of states S. Decision problems are conveniently represented by decision trees as illustrated in Figure 3.10, where each node represents a decision or action taken out of a space of possible decisions or actions.

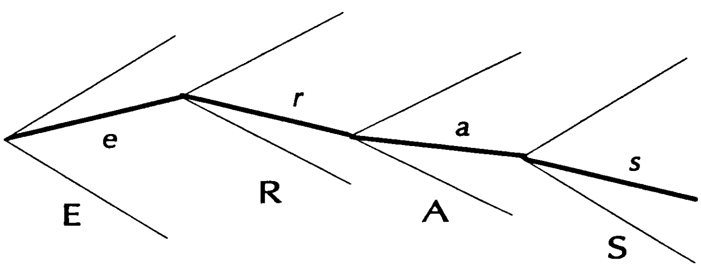


Figure 3.10 Decision tree

The formulation of a decision problem in this way allows the use of a Bayesian approach to evaluating probabilities. This approach uses subjective information and has been applied to IRM planning as outlined below. Fuller explanations of the basis for Bayesian decision analysis can be found in Raiffa and Schlaifer (1960), Benjamin and Cornell (1970) and Ang and Tang (1984).

In inspection scheduling, the experiments E correspond to possible inspection techniques and inspection times; the results R to inspection results, that is, no cracks found or the sizes of detected cracks; the actions A correspond to setting a repair criterion for carrying out welding or grinding; and the state S to the reliability of the tubular joint. The performed experiment and the chosen action, with the result of the experiment and resulting reliability of the component, determine the utility value. The measure of utility could, in IRM planning, correspond to overall costs.

To perform decision analysis, it is necessary to have information on the experiments E and actions A, i.e. the possible inspection techniques combined with inspection times, and the repair actions, with the utility function $u(e,r,a,s)$, or alternatively, some form of cost function $c(e,r,a,s)$. The only difference between $u(.)$ and $c(.)$ is that whereas a high utility value is desirable, a low value for $c(.)$ is preferred.

In addition, information on the probabilities of various combinations of possibilities is required. The joint probability $P_{S,R}(s, r|e)$ that the universe is in state s and the result of the experiment is r, given that experiment e was carried out, is required. In other words, some understanding of how the failure of the joint is related to an inspection result is necessary. $P_{S,R}(s, r|e)$ will also give four important probabilities. One is $P_S(s)$, representing the cracks which may possibly exist irrespective of previous inspections or repairs. $P_S(s)$ is a *prior* probability since a value is assigned to this prior to knowing the results r of the experiment. Another is the conditional probability $P_R(r|s,e)$ for a particular result obtained from an experiment, that is, the probability of an inspection result, given the existing state of the tubular joint and the experiments carried out. $P_R(r|s,e)$ is a measure of the reliability of the inspection technique. Also required is the conditional probability $P_R(r|e)$ for a result r given experiment e; or given an inspection, the probability of an inspection result being r irrespective of the state of the tubular joint.

Since the above information is what is required to apply Bayes' rule (see Equation 3.17), the probability of the state of the universe or of the failure of the tubular joint, given inspection results and repair actions, can be obtained by

$$P_S(s|r) = \frac{P_R(r|s,e)P_S(s)}{\sum_S P_R(r|s,e)P_S(s)} \tag{3.25}$$

In summary, the decision problem is to choose the experiment e, given E, R, A, S and $P_{S,R}(s, r|e)$, such that the cost function $u(.)$ is minimised (Raiffa & Schlaifer, 1960). For IRM planning then, this choice of an experiment corresponds to a choice of an inspection technique and time, such that the expected costs are minimised.

One way of carrying out the analysis which will result in a minimum cost value is to employ normal form analysis. In this, a decision rule *d* is specified which prescribes the action which must be taken for all possible outcomes of the experiment *e*. For each experiment *e* an optimal decision rule can be found. Thus, by considering all possible experiments e, the overall optimal experiment can be selected. The decision rule for a specific experiment e is a mapping d(.) from R into A, that is, the rule assigns a specific combination of repair and inspection action d(*r*) that must be carried out given a specific inspection result r for a specific technique.

For an experiment e, the expected cost is given by

$$c(e,d)=E_{S,R|e}[c(e,r,d(r),s)] \tag{3.26}$$

And the optimal experiment *e* and the optimal decision rule can now be identified by solving

$$C=\min_{e}[\min_{d}[c(e,d)]] \tag{3.27}$$

3.3.2.2 Identification of Optimal IRM Plans

The optimal inspection and maintenance plan is the plan minimising the total cost *c*(.) associated with the inspection and maintenance of the structure throughout its anticipated lifetime. In general, the total cost *c*(.) must include the costs associated with the inspection, with repairs, and with the event of failure which in turn may include costs associated with loss of production. The expected total cost, associated with a particular inspection and maintenance plan carried out at a point in time *t*, is given by

$$E[c(t)]=E[c_s]+E[c_a]+E[c_e] \tag{3.28}$$

where c_s is the cost of the state of the universe or for this case, c_s = cost of failure c_f, $c_a = c_r$ is the cost of the repair action and $c_e = c_i$ is the cost of inspection.

In the analysis carried out for RISC, minimisation is not performed. Instead, the total expected cost $E_{S,R|e}[c(e,r,d(r),s)]$ corresponding to a specific inspection *e* and maintenance *d(e)* plan is estimated.

3.3.3 Modelling of Inspection and Repair

To obtain the total expected costs for a plan of action on a joint, requires the definition of the events and the associated costs. The relevant events are the past inspection and one future inspection combined with repair.

3.3.3.1 Inspection Events

For the two inspection events, the different possible situations that arise need to be identified. When considering previous inspections, the possible results or events are

1. No crack is detected.

2. A crack is observed with a depth equal to A_{obs}.

3. A crack is observed with a depth smaller than A_{obs}.

4. A crack is observed with a depth larger than A_{obs}.

For practical purposes, only events 1 and 2 are considered, since it is assumed that as much information as possible will be derived from the inspection.

Slightly different outcomes are possible with the future planned inspection. Ignoring the events that are equivalent to events 3 and 4 above and adding new events related to the possibility of failure before the next inspection, they are,

1. No crack is detected.

2. A crack is observed of depth equal to A_{obs}.

3. The crack depth at planned inspection is smaller than the critical crack depth.

4. The crack depth at planned inspection is larger than the critical crack depth, which implies failure before the next inspection could take place.

The probability for each of these events can be evaluated by defining state functions in a similar way to the failure state function. The exact probability is given by the integral of the joint probability density functions over the space of the basic variables, where the state function takes on the appropriate value. Structural reliability analysis techniques can be applied to find a measure equivalent to the reliability index, which is then used to obtain an approximation to the actual joint probability.

The quality of inspection, its sensitivity and accuracy, is modelled by a distribution $f_{Ad}(a)$ for the detectable crack depth A_d, corresponding in some form to the probability of detection (POD), and by a second distribution $f_\epsilon(e)$ for measurement uncertainty ϵ_{insp}, corresponding to probability of sizing (POS). When cracks are not detected, then the POD measure is used to update the information on the crack depth. For observed cracks, the POS measure is used instead. The distribution form and the parameters for the POD and POS measures differ from inspection method to inspection method.

3.3.3.2 Modelling of Repairs

Having identified and measured a crack at the time of inspection, a decision must be taken with respect to possible repair actions. The decision is usually well defined, in that some size is given so that if a crack smaller than the prescribed size is found, then no major repair takes place, otherwise, a repair will be carried out.

Preliminary grinding of the crack is very often performed before actual sizing of a defect. This is to avoid the situation that a surface crack with a very small depth is measured using an expensive sizing technique. If the crack is still present after the preliminary grinding, rigorous sizing is performed and after this, a decision is made on the actual repair action. Thus, in effect small cracks are removed by grinding. So, two types of repair actions are considered: grinding for small cracks and welding for larger cracks. In addition, it is possible that failure occurs after repair. The corresponding events are

1. If the crack depth is smaller than A_{weld}, the crack is ground away.
2. Otherwise, if the crack depth is greater than A_{weld}, then weld repair takes place.
3. The crack depth at the end of the service life is smaller than the critical crack depth, after repair.
4. The crack depth is greater than the critical crack depth, after repair and before the end of the service life, which implies failure after repair.

The first two possibilities correspond to the events,

1. A crack is observed with depth greater than A_{weld}.
2. A crack is observed with depth smaller than A_{weld}.

As before, the probabilities of these events can be relatively easily evaluated by using POS information. The value for A_{weld} is set for each joint individually, based on its past in-service history. A_{weld} has a lower value for a joint that has been ground in the past, as compared to a joint that has not been ground at all.

Material properties after grind repair will be the same as before grind repair. The initial crack depth and length after grind repair can, in general, be assumed to be smaller than those assumed at the start of the lifetime of the joint. A change in the initial crack is taken into account by considering random variables representing initial crack depth and crack length after grind repair, which are independent of those used to model the initial crack depth and length at the start of the lifetime of the joint. Similarly, as the tubular wall thickness is changed during grind repair, the random variable modelling tubular wall thickness after grind repair is also assumed to be independent of the tubular wall thickness random variable at the start of the lifetime of the joint.

After weld repair, it is usual to use the same model as at the start of the lifetime, both with respect to initial crack geometry and with respect to tubular wall thickness. Since crack growth behaviour after repair is independent of that before repair and to allow possible new models, perhaps to reflect the fact that underwater welding is not as precise as welding at manufacture, new random variables are introduced for the crack depth and wall thickness after weld repairs. Similarly, new random variables representing the Paris Law parameter C after welding are introduced, as material properties are expected to change after welding.

3.3.3.3 Calculation of Costs and Sensitivities

As explained in the previous sections, the optimal inspection and maintenance strategy can be identified by estimating and comparing the expected total costs corresponding to the possible options of inspection types, inspection times and repair actions. According to (3.28), this requires calculation of the expected failure costs, the expected repair costs and the inspection costs. The total expected cost of failure has to account for failure before inspection has taken place and after inspection, that is, between the time of inspection and the lifetime of the structure:

$$E[c_f] = P_f(t_{insp})C_f + P_f(T_L - t_{insp})C_j \qquad (3.29)$$

where C_f is the cost of failure, $P_f(t)$ is the probability of failure occurring in the interval of time t assuming no failure before the start of the interval, T_L is the lifetime of the structure and t_{insp} is the time to the next inspection. To calculate the probability of failure occurring between the previous inspection and the next inspection point, that is $P_f(t_{insp})$, requires considering only the possibility that the crack be larger than the critical crack size before the end of the service life, given the previous inspection results. To calculate the probability of failure occurring after the next inspection, $P_f(T_L - t_{insp})$, requires considering the all possible combinations of the crack growing to the critical crack size and the inspection outcomes, given the previous inspection results.

Similarly, the expected cost of repair $E[c_r]$ is calculated as

$$E[c_r] = P_{grind}(t_{insp})C_{grind} + P_{weld}(t_{insp})C_{weld} \qquad (3.30)$$

where $P_{grind}(t)$ and $P_{weld}(t)$ is the probability of grind or weld repairs, respectively, being required immediately after inspection at the inspection point t, and where C_{grind} and C_{weld} are the costs of grinding and welding.

The cost of inspection C_i is a constant in that the inspection will take place, even if failure has occurred. The total expected costs are given by

$$E[c_t]=E[c_f]+E[c_r]+C_i \tag{3.31}$$

Typical results for expected costs calculated for t_{insp} over the interval $[t_{insp}*, T_L]$, where $t_{insp}*$ is the last inspection point, are shown in Figure 3.11. Ti is the optimal inspection time which leads to minimal expected costs.

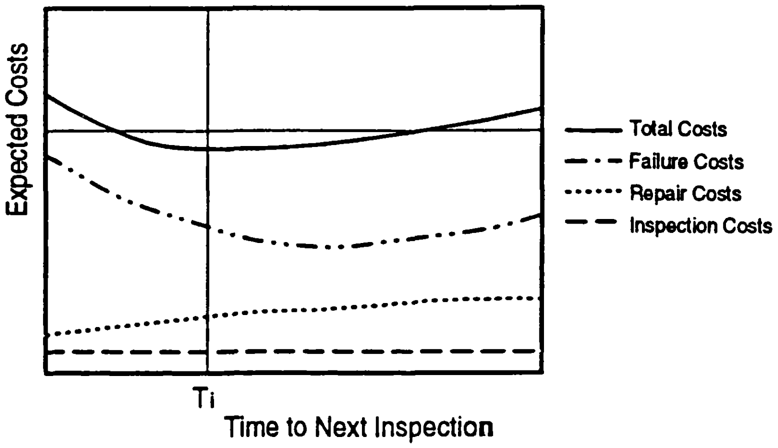


Figure 3.11 Expected costs

Inspection and maintenance actions are, in effect, long term investments, so the real rate of interest should be taken into account. The calculations for this are standard and were included in the RISC project, but are not discussed here.

3.3.4 Evaluation of Updated Reliability

To be assured that the reliability of the considered welded connection is acceptable, it is necessary to provide estimates of the reliability index β as function of the current age or lifetime t of the structure, in addition to the expected costs.

If necessary and in addition, the POF associated with an inspection and maintenance plan can be estimated easily, by making use of the $P_f(t_{insp})$ and $P_f(T_L - t_{insp})$ already calculated:

$$P_f = P_f(t_{insp}) + P_f(T_L - t_{insp}) \tag{3.32}$$

This total POF can also be used for deciding which inspection plan to carry out, by selecting the plan with the smallest POF.

If the considered joint is inspected at time $t_{insp}*$, the observations from this inspection can used to obtain $\beta(t)$, the reliability index, at points in time t in the interval $[t_{insp}*, T_L]$. The updated values at each point are calculated using of the relationship between β and the POF. A graph of typical results of updating β by inspections at time T_1, T_2 and T_3 is shown in Figure 3.12.

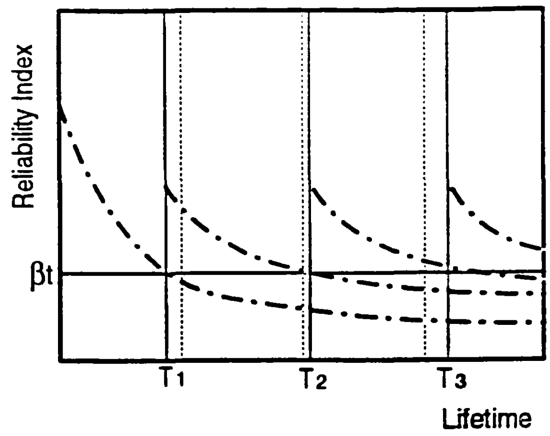


Figure 3.12 Typical updated reliability index values

3.3.4.1 Use of Inspection Results

Once an inspection on a component has been carried out, the results have to be used in some way to update our information on the state of the component. Values for the reliability of the component can be recalculated by carrying out the analysis as if from the initial state, or, as discussed above by making use of the more rapid Bayesian updating method. The latter method is employed in the RISC methodology.

The uncertainties in the inspection need to be incorporated in to the analysis. Indeed either method requires data on the reliability of the technique, in the form of POD as a distribution of detectable crack sizes and as a POS. Two scenarios may be considered: either no crack is detected or a crack has been detected and sized. The case that a crack is detected but not sized need not be considered, since this would not arise in practice: any defect found is normally reinspected to confirm its existence and sizing would then be carried out.

3.3.5 Structural Analysis in RISC

The reliability analysis carried out for the RISC procedure is based on the description given on fatigue crack growth modelling, load modelling and optimal scheduling. The complete procedure requires several stages and was implemented as a small set of linked software modules. These are shown diagrammatically in Figure 3.13. The stages are

1. Loading analysis carried out by the program ULDAN to provide stress history data.

2. Component level reliability analysis carried out by the package RISCREL based on Level II structural reliability methods FORM and SORM.

3. Within the RISCREL package lies a limit state module based on fatigue fracture mechanics and employing fast assessment techniques.

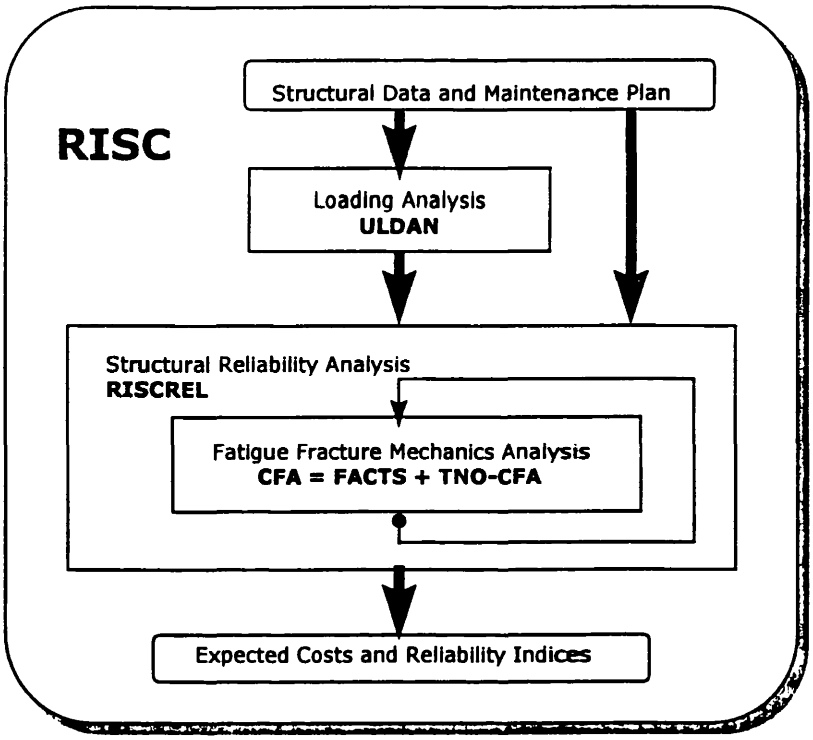


Figure 3.13 Structural analysis in RISC

3.3.5.1 Loading Analysis with ULDAN

As the fatigue loading experienced by the offshore platform is an important input for the crack growth calculations, it is important to analyse loading reasonably accurately. There are several considerations dictating the ways in which loading analysis can be carried out.

The first issue is the method of load analysis. A fundamental method is to model the structure as a frame, that is, a system of connected beams, and carry out finite element analysis with the wave data as input to find the load on each connection. To do this within the reliability analysis is of course too time-consuming. To do this outside the reliability analysis does not provide results better than other more analytical approaches, such as spectral analysis. In addition, for some older structures only deterministic data as exceedance curves has been stored as information on the loads and thus sophisticated analysis is not possible. In the RISC project, it was decided that the entry point for the reliability analysis was predefined loads for each joint. The preferred method is to produce a stress range probability distribution (SRPD) from sea states information and stress distributions for each point on the joint. An exceedance curve. modelled as a 2-parameter Weibull distribution, can be used to describe the SRPD. The ULDAN module carries out the SRPD analysis and the procedure is shown in Figure 3.14.

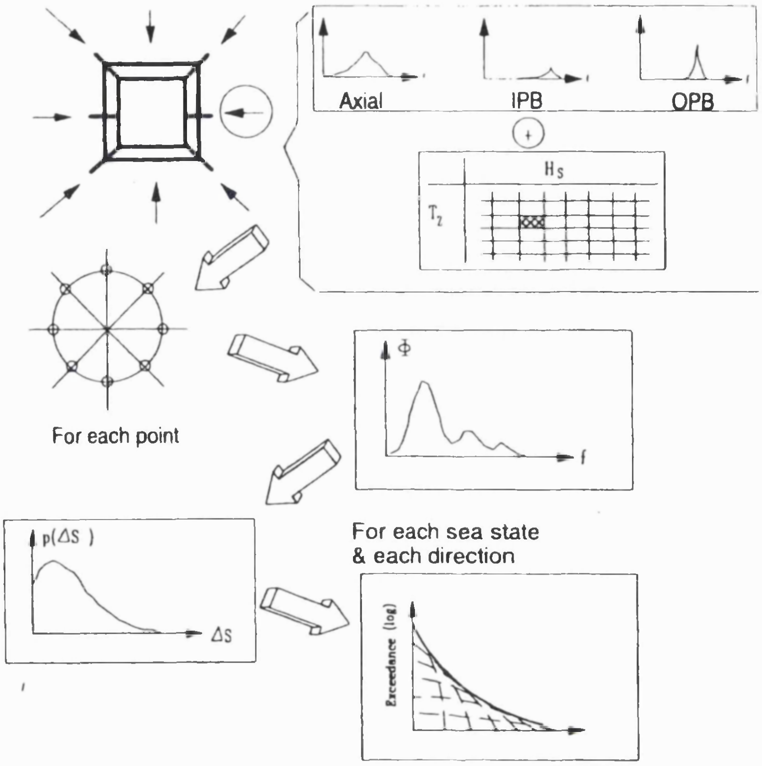


Figure 3.14 The loading analysis procedure

Another issue is where in the analysis procedure should the loading analysis be carried out. Loading analysis could be carried out within the reliability analysis algorithm, that is, it could be included in some way in the failure function. Limit state parameters corresponding to the dynamic environmental loading on the structure would then be input to the reliability analysis. These parameters would then have to be related to the loading at the joint in question at each iteration. This approach would increase the computational effort to a level beyond that which is feasible. At the component level, however, it would not provide much better results. The more pragmatic approach is to provide the limit state with parameters or random variables corresponding to the loading at the connection.

3.3.5.2 Fatigue Fracture Mechanics Module (CFA)

The overall fatigue crack growth analysis is shown in Figure 3.15. The fracture mechanics modelling as described in previous sections has been implemented as a computer module known as Component Fatigue Analysis (CFA). The CFA module is a combination of the FACTS program developed at UCL (Dharmavasan, 1991; TSC, 1990) and the modified FAFRAM program, TNO-CFA, of TNO (Dijkstra & Straalen, 1991). This module calculates a predicted crack size at a point in time by integration of Paris' crack growth model, given loading data in the form of the SCF and the SRPD, the material properties C and m, and initial crack size and critical or final crack size.

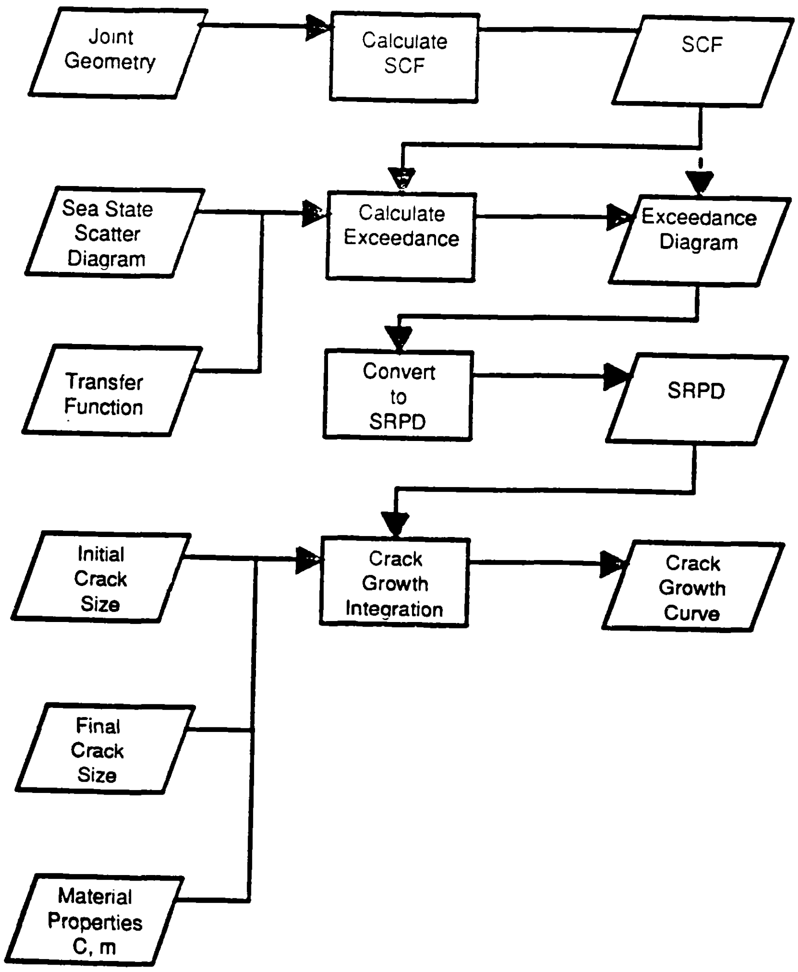


Figure 3.15 The overall crack growth analysis procedure

The CFA has a number of crack growth relations available and the most appropriate method used for the reliability analysis can be chosen as required. A more detailed description can be found in a paper on the CFA (Dijkstra et al, 1994).

3.3.5.3 RISCREL

The reliability analysis using the fatigue failure criterion described was implemented by programmers as a set of FORTRAN modules. The overall output and input for RISCREL is shown in Figure 3.16.

The RISCREL program also carries out Bayesian updating based on a past inspection result and cost evaluation of a maintenance plan for a joint. RISCREL takes as input probabilistic models for the initial crack geometry, the joint geometry, material characteristics, in addition to the long term statistics of the stress processes which are specific to the hot-spot. The statistical data on the stresses and the SCF are stored in a joint file and the material properties are held in a database. The costs required are on failure of the component, of the inspection and of repair. Other data required includes the analysis options, such as stress intensity factor solution, and inspection reliability for used and planned techniques.

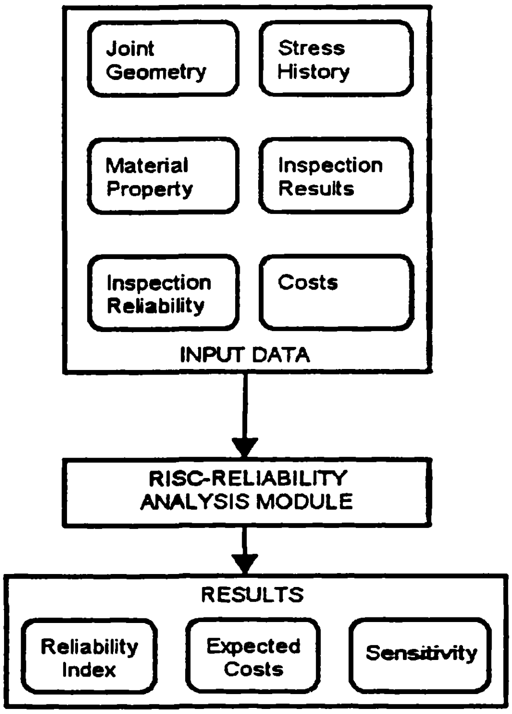


Figure 3.16 The reliability analysis module, RISCREL

To allow the analysis module to be embedded within any system, the communication, both input and output, by text files. The output consists of updated reliability indices or expected cost estimates. More detailed results such as partial derivatives, information on algorithm monitoring, etc. are also possible.

3.4 SUMMARY

Jacket type structures are highly redundant structures subject to many uncertainties and in particular, to time-varying, cyclic loads and so to fatigue. Conventional reliability assessment of these structures is not always feasible. Instead, this chapter described the use of reliability analysis techniques recently developed specifically for structures, based on simplifications of the functions describing failure.

The application of reliability methods in the design and maintenance of offshore structures has only been developed recently. For many reasons, such as the degradation of existing structures and the requirement to be able to extend the life of some structures, there is a need to be able to analyse the reliability of these structures in a more rigorous manner.

The approach adopted was to provide a tool that would estimate the expected total costs associated with a particular inspection and maintenance plan. In that way, different inspection and maintenance plans can be compared, constraints enforced and from this an appropriate rational plan chosen.

The analytical tools described in this chapter formed the basis for the RISC methodology. The assumptions employed in their design were based on the survey of current procedures provided by the author and described in Chapter 2. They were implemented as the software modules ULDAN and RISCREL, incorporating the CFA module, which lie at the heart of the RISC System.

The real problem lies in applying the reliability-based analysis to the production of rational inspection strategies. To use reliability analysis requires understanding the inspection procedure and making use of the appropriate information on inspection techniques. The problems of generating input data for the analysis and of interpretation of the results remain. The interpreted results must then be combined to form a rational global solution for the structure that takes into account constraints which may not have been included, implicitly or explicitly, into the analysis. It is necessary to consider developing complete systems which will aid in carrying out the analysis and interpreting the results.

The complete RISC System integrates the analysis modules within a framework that controls execution of the analysis modules. A clear understanding of the theory and the assumptions made in formulating the analysis methodology was required by the author in order to design and implement the system described in Chapter 5. The RISC Demonstrator includes a module that generates suitable schedules from the results, based on the information given in Chapter 2 and in this chapter. The following chapter reviews techniques from the artificial intelligence field which were considered for use for the complete RISC System.

4 KNOWLEDGE BASE SYSTEMS FOR IRM PLANNING

A computer system for improved IRM scheduling for fixed offshore platforms is required that uses various forms of information to generate suitable schedules. Such a system will need to be able to process and utilise correctly all the required data and information for structural reliability analysis and the fatigue fracture mechanics analysis. As these techniques are relatively new and there is often little documentation on them. Additionally, these techniques require many different forms of data from many different sources.

It was proposed that a knowledge base system (KBS) for RISC would be able to tackle the requirements of

- flexibility of the representation scheme for the different types of information
- effective handling of incomplete information on the structure
- selective utilisation of knowledge for efficiency
- the capability for operators to control the problem solving sequence
- ease of maintenance and extension as new information is gathered and as inspection procedures change

This chapter reviews the background to artificial intelligence (AI), describes the basics of knowledge base systems and their workings and defines some of the most common terms. In addition, general planning and scheduling issues and algorithms are reviewed and described.

4.1 REVIEW OF ARTIFICIAL INTELLIGENCE CONCEPTS

Artificial intelligence is the study of human reasoning (Winston, 1977; Charniak & McDermott, 1985). There are two aims in AI:

- to understand the human mind through simulations making use of computers
- to make computers more intelligent and hence more usable

These two goals also represent the dichotomy which exists in AI. On the one hand, the researchers whose main interests lie in psychology investigate theoretical models of the human reasoning process and the systems produced may be very narrow in application. On the other, applied computer scientists and engineers apply AI concepts in the most pragmatic way to provide practical systems which help in decision making. The applications that are considered in AI include memory organisation, natural language processing, understanding and reasoning, which are areas of particular interest to the psychologists; and vision, machine learning or cybernetics, control or robotics, and, of course, planning and scheduling.

Artificial intelligence was given its name by John McCarthy in 1956 in the proposal to run a workshop meeting at Dartmouth College, US, to review what appeared to be a new and promising field of research (Charniak & McDermott, 1985). AI has its roots in the thought experiments carried out by Turing on the intelligence of computers, the chess-playing computer by Shannon and in the theory of logic invented in the previous century by Boole. It was McCarthy, Minsky, Newell and Herbert who first started working directly in this field.

Very early work in AI was based on the idea that humans reason at a high level with abstractions and carry out symbolic manipulation. By this it is meant that if someone was asked why they choose to do something a certain way, they would be able to provide a description of their reasoning process, as described in work on a first AI system, General Problem Solver (GPS) (Newell & Simon, 1963). This work led to the development of rule-based systems, which were generalised over time to encompass what are now known as expert or knowledge base systems. The representation of knowledge for these systems was soon to be recognised as a major issue, as the first formal discussion by McCarthy (1968) indicates.

By the 1970s, it was realised that most human reasoning is at a much lower level. For instance, humans cannot always explain how they have recognised that the shape of plotted sensor data indicates a crack, or why they have chosen to carry out a repair immediately. The use of logic with its requirement for consistency in inference, independent knowledge modules (propositions and facts) and the separation of knowledge from the inferencing procedures, may be inadequate to explain all human reasoning (Minsky, 1972; Dreyfuss, 1981). This realisation led to work in alternative forms of knowledge representation such as frames, but also in areas such as neural nets for pattern recognition that attempt to model the physical processes occurring within the brain.

Other related areas of computer science include genetic programming or the use of genetic algorithms. In a genetic algorithm, a population of individuals, corresponding to a subset of possible solutions to a problem, combines to form new populations or new solutions. The new solutions are evaluated and the 'fittest' are allowed to combine to create another generation of solutions and so on, until a very good solution is found (Davis, 1987; Bolc & Cytowski, 1992). Thus to be applied to scheduling, a genetic algorithm requires an initial population of possible schedules. For RISC part of the problem is to find an initial schedule. Thus genetic algorithms were not considered in detail for this work and so are not explained further.

The basis for neural nets and pattern recognition is described briefly below. As KBS are of much more interest to the work in RISC, these are discussed in more detail.

4.1.1 Pattern Recognition Techniques

The term *pattern recognition* is almost self-explanatory. Pattern recognition involves the extraction of features from a large set of data and is applied traditionally to problems of vision and, to a lesser extent, hearing (Elliman & Banks, 1989).

In early AI work most human reasoning problems, other than the very specific vision problem, were considered to be solvable by the application of high level reasoning and logic. It is now understood that in many situations humans develop an intuition about how to solve a problem: something about the problem stated makes them think that perhaps such a solution would be appropriate. The "something" is undefinable and the "perhaps" only emphasises that there are uncertainties in the reasoning. Forcing explicit representation of the reasoning process, however, would lead to fundamental difficulties later on as inconsistencies become evident, in that solutions for new situations are not agreed upon by the explicit reasoning process and the human. For this reason, this type of problem is now treated by methods which are to a certain extent *black box* techniques, that is, the inner workings are unknown by the user, only the input and output are of interest.

4.1.1.1 Issues of Pattern Recognition

Pattern recognition involves selecting case data which is relevant to the problem, identifying or extracting features with which to allow different results to be represented and then formulating an analytical method that gives values for the features when presented with a new case (Elliman & Banks, 1989). As an example, in the inspection data interpretation problem for a sensor employing the ACFM technique (explained later in Chapter 7), the data may be the magnetic field data from the sensor, the features may be the characteristic peaks and troughs which may be used to identify different types of defects, and the method is required to identify the peaks and troughs from the mass of magnetic field data in some way. As will be described in Chapter 7, the method for this pattern recognition process was a simple local minima and maxima searching procedure, which did not work well in practice.

There are many techniques which carry out pattern matching. Some are very specific to the application, for instance signal processing work is reliant on mathematical transforms of wave data. General techniques include classical statistical model fitting, clustering techniques, neural networks and to a certain extent fuzzy systems. The best-understood of these are the classical statistical methods, but these require a model to be stated and the data is fitted to this model. Clustering techniques or discriminant analysis can also be problematic in that the data has to form distinct clusters in an n-dimensional space of the variables (Elliman & Banks, 1989).

There is much argument over whether or not neural networks really do provide a new technique, as is believed by the AI school of thought, or if they are just another clustering technique, as the statisticians believe (Kosko, 1992). The AI school believe that in fact neural nets and fuzzy systems are equivalent. Fuzzy systems are explained in Section 4.2.3.2, after neural nets are described.

4.1.1.2 Neural Networks

Based on the first aim of AI, much work is now being carried out on simulating the human learning process at neuron level, that is trying to model what is thought to occur within the brain of a human (Strange, 1989). This work led to the concept of neural networks, also known as artificial neural networks (ANNs), which are adaptive systems responding to stimuli (Kirk & Lewcock, 1995; Kosko, 1992). Neural networks have the very useful property of being able to recognize patterns that are difficult or even impossible to define, that is, no model is required before presenting data to a neural network, unlike classical statistical techniques. It is outside the scope of this thesis to discuss neural networks in detail, although the basic concepts are outlined.

A neural network can be visualised as a directed graph, that is, as a set of linked nodes, where the order of the nodes matters. The nodes are in the input, output or hidden or intermediate layers. Signals or data are received by the input nodes; this affects the nodes in the hidden layers, and hidden nodes in turn activate the output nodes which send out the data from the network. The links are weighted and the hidden nodes take on a state which is dependent on the state of other nodes, usually only the nodes in one layer closer to the input layer, and the weights of the links. An example of a forward feed neural net is shown in Figure 4.1 with three hidden layers of three nodes each. This is a typical network in that the number of hidden layers is usually greater than 1, and the number of input nodes is usually greater than the number of output nodes.

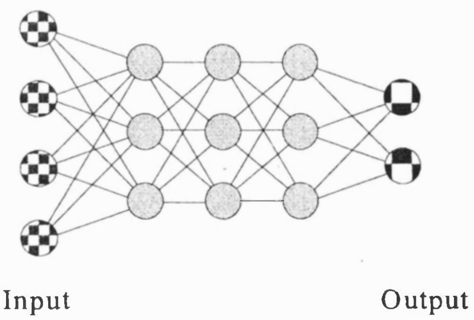


Figure 4.1 A simple neural network

As an example of a possible neural network, in the ACFM signal recognition problem mentioned earlier, a neural net was proposed by DASA in Munich to help with the characterisation of defects from ACFM sensor data (Ellingworth et al, 1992). In one proposal, the neural network had many input nodes, one for each point on a matrix of 2-D coordinates on a surface and the data at each input node was the average of nine magnetic field values around the point. The output nodes were only a few, one for each defect type.

The simplest form of the relationship between the nodes is a weighted sum of the states of the nodes in the layer before it. So for the ith node in the jth layer, its state a_{ij} is given by

$$a_{ij}=\sum_{r=1}^{M} w_{irj}a_{r(j-1)} \tag{4.1}$$

where a_{i0} is the given input layer, $0 < j \leq N$, j=N represents the output layer; w_{irj} is the weight of the link between the rth node in the (j-1)th layer, that is the node with state $a_{r(j-1)}$, and the ith node in the jth layer; and M is the maximum number of nodes in any one layer. The simplest neural net is one in which the nodes can only take a value from the possible values {0,1}, where 0 represents an inactive node and 1 is activation. Alternatively another set of possible values could be {-1,0,1}, where -1 represents inhibition, 0 non-activation and 1 activation. The sum above would then be modified by including a threshold rule, such that if the sum is greater than 0.5 say, then state a_{ij}=1, and 0 otherwise. The weights w_{irj} can be inhibitory, that is be negative in value; have no effect, that is zero; or stimulative, that is, have a positive value.

The strength of neural network theory is that it includes a learning algorithm. The weights between each node need not be defined. By presenting suitable training sets of data, that is to say, sample input and with the required output, iterative methods exist to determine the weights such that the required output is given from the input. What has to be predefined is the architecture for the network, that is, the form of the input data, the number of hidden nodes and layers, the output data format and the form relationship between the nodes.

The strength of this technique is also its weakness. The iterative search for a set of weights can be non-convergent, that is, may never come to a solution. It has been found that this process is highly sensitive to the network architecture chosen. For instance, changing the number of hidden layers by one, or varying the input layer, or selecting slightly different output features, may impact greatly on the convergence of the training process. A neural network which could be successfully trained, may, with slight variations, become a non-successful neural network, or vice versa. Training is also affected, but not as severely, by the choice of training set. Enough sets of input and output data must be chosen to ensure that the network will recognise new sets. On the other hand, too many training sets can make the resulting neural net over-sensitive and unable to recognise new sets. Unfortunately, there is no way at the moment of predicting when training will succeed. Fortunately, when it does succeed, the resulting neural network becomes a very powerful tool for rapid pattern recognition.

4.1.2 Expert Systems and Knowledge Base Systems

There are various definitions, which do not all coincide, of what is meant by an *expert system*. One view is that it is a program which demonstrates expertise, that is carries out a task in the same way, seemingly, that a human expert would (Johnson & Keravnou, 1985). Another view is that it is a program in which the knowledge on how to solve a task is separate from the processes required to manipulate that knowledge (Alty & Coombs, 1984). In this thesis, the latter definition is reserved for *knowledge base systems* and an expert system is considered to be a specialised knowledge base system, one which carries out a task expertly.

The first definition of an expert system was based on the concept of "a classroom of shrieking daemons[1]" proposed by Newell (Winston, 1977). He visualised human reasoning within the brain to be akin to a problem being solved line by line on a blackboard in view of many naive, simple solvers, or daemons, each of which knows how to go from one line to the next, but only for one step. It may be that for some steps, there are several daemons that believe they know how to take the next step. This is not a problem as a teacher or monitor can choose one of the daemons currently indicating, by shrieking perhaps, that they can solve the next step, to come up to the blackboard and carry out that step. If at any point there is no daemon to contribute a next step, then the monitor backtracks to a point in the problem solving process where there was more than one shrieking daemon and chooses another daemon, in the hope that this daemon will lead closer to the solution. Alternatively, if parallel computation is possible, then all daemons can be asked to contribute to the blackboard at the same time. Several hypothetical solutions would be investigated in parallel and, assuming infinite parallelism, no backtracking is required. This became the first model for expert systems and from this, for knowledge base systems.

This naive model emphasises the idea that the problem-solving process involves many chunks of simple knowledge, and that the knowledge is separate from the control of the process. Thus, the architecture of a KBS is usually defined as comprising three parts:

4.1.2.1 Knowledge Bases

The knowledge bases contain the information that is common to a type of problem: the heuristics used, classes of components or concepts and procedures. The knowledge base is the most problem-specific part of the system. The term is really a generalisation of the much better known term database. Unlike a database where a very specific type of information is kept in the form of

[1]The spelling here is of significance as the most usual meaning applied to "demon" is that it is an evil spirit or devil, while "daemon" is more usually associated with a guardian spirit. It is hoped that these "shrieking daemons" are more benevolent than malicious.

records, a knowledge base could and often does contain all types of information. In practice, a KBS does not usually have a single knowledge base. Knowledge is separated into modules wherever possible for practical reasons. First of all, modularising enables the easy re-use of knowledge bases (Sticklen et al, 1987), for instance, knowledge of scheduling in the RISC System may be used in another scheduling KBS for another type of structure or another industry. Also each knowledge base may be loaded when needed to reduce memory requirements. Conceptually, it is irrelevant whether there is only one large knowledge base or several small distinct knowledge bases.

The main problem in the development of a KBS is in deciding on what knowledge representation scheme to use to construct the knowledge bases and how it should be implemented.

4.1.2.2 User-Interface

In common with any complex computer program, a suitable man-machine interface or user-interface is required for any expert system. A user-interface will need to provide explanation facilities, prompt for input, let the user to control the reasoning process whenever possible and display results. In theory it is possible to have a semi-natural language interface, that is an interface which would understand certain key phrases and output English-like statements. An example of this type of dialogue could be:

User	`Show me why the length is 10 cm.`
Machine	`The length of the structure BLOCK is 10 cm as a result of running analysis PROG.EXE with datafile MYDATA.DAT`

What is not practically possible is to be able to produce an interface which could understand all forms of input. In practice then, the interface is expected to deal with only a minimal vocabulary appropriate to the application and to the level of users.

The display of results is of course also important. Engineering applications nearly always best display data through the use of pictures, graphs and diagrams. The user should be able to interrogate the values for output parameters as and when results are achieved, that is derived, calculated, etc., and if appropriate change the route followed by the system.

The possible forms of interaction between a user and a computer system are initiation by the user or by the computer (Johnson & Keravnou, 1988). In the user-initiated mode of interaction, the user decides, based on the computer output, to proceed with the inference and provides the computer with the route to follow. In the computer-initiated mode, the computer alone decides, based on the user-input, to continue deriving more accurate conclusions and asks the user for the appropriate data. The most natural form of interaction is a mixed mode. Additionally, for engineering applications it is now not normally expected that a knowledge base system replace the engineer; instead it is expected that it act as a consultation system providing aid to the users as required, with the engineer making the final decision at each stage.

Apart from explanation facilities other important facilities are that the user be allowed to interrupt and store sessions, and produce reports. These allow time to find needed information that was not at hand at the time and provide records for later inspection.

4.1.2.3 Inference Engine

In the same way that a human expert is required to reason with and deduce new information, a KBS has an inference engine to reason with the contents of the knowledge base and with the input data. Another description is that the inference engine is the program whilst the knowledge base and input is the data required by the program. In practice the distinction between inference engine, user-interface and knowledge base is not at all clear cut. It is not always possible to separate out the inference engine from the problem-specific knowledge and the user interface would make use of information in the knowledge base. As an example of this, consider the item

```
If input is not comprehensible
then give a message to the user "Sorry, I only
understand English." and stop.
```

The developer has to decide where best to include this information and the decision is made within the context of the application. Mistakes made at this point severely hamper future modifications made to the KBS.

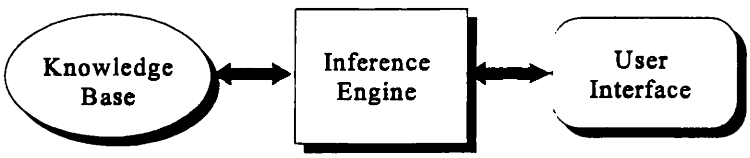


Figure 4.2 Components of a KBS

The above gives a macro-level description of a traditional knowledge base system. It is necessary to consider more closely the nature of each part of a knowledge base system.

4.2 ISSUES IN KNOWLEDGE BASE SYSTEMS

The problem which most frequently arises in KBS development is that of deciding what knowledge is required and how to select an appropriate knowledge representation scheme. The different formalisms which have been defined in the field of AI are explained in this section.

The next problem is that of choosing a reasoning scheme, which is often tied to the representation. Almost any form of reasoning involves searches at some point or another. Reasoning techniques are usually closely associated with a particular knowledge representation scheme and are outlined here. Searching techniques are also employed explicitly in the planning problem and hence these are described later on in chapter in greater detail.

4.2.1 Knowledge Representation Schemes

One of the main distinguishing features of knowledge base systems is that the problem-solving information used by the system is made explicit. This only becomes an advantage if the language and grammar used to represent this information is appropriate for its type, so that it can be represented in a natural form, by which it is meant a form which closely matches an expert's own definition and is also sufficiently complex to allow the representation of complex relationships. Many different formalisms have been proposed and this section describes these formalisms and explain their advantages and disadvantages. First, it is worthwhile attempting to define the different levels or layers of information required by any KBS.

4.2.1.1 Layers of Knowledge

For practical reasons, it is necessary to distinguish between *static* and *dynamic* knowledge. *Static knowledge* is the information used by a software system which will not change in the normal course of consultation with the system. For example, the inspection reliability databases used by the RISC system would not be changed whilst carrying out reliability analysis of a component which makes use of an inspection result. *Dynamic knowledge* is the information that is particular to the current session, for instance input data or results obtained from analyses. It should be clear that this is a fairly artificial distinction between the two types of data. For example, the data in the inspection reliability database would change as more information is gathered on the inspection technique, and certainly much of the dynamic knowledge on the structure would be stored as a history of the structure and hence become static knowledge. It is convenient, nevertheless, to view the two types as distinct as their requirements do vary slightly as described below.

The basic and practical requirements for a knowledge representation scheme are that it should be easily accessible by the software system; understandable in that it is a close fit to language in which the problem is described; and finally expressive (Fikes & Kehler. 1985). By *expressive*, it is meant that the scheme must have a generalised format which can accommodate various types of information, such as

- documentation for the static knowledge
- heuristics that aid in narrowing down in the search for a solution, that is "rules-of-thumb"
- algorithms and procedures
- classes of objects to describe, for example, components or materials
- information on the dynamic data, for example, at what point of the consultation it was computed and how it was derived
- graphs and diagrams for input, output and explanation

More abstractly, the following layers of information for a KBS were proposed by Bobrow and Winograd, (1978):

- task domain which is knowledge specific to a problem, for instance about a fixed platform, or reliability analysis
- interaction domain refers to knowledge about the language for communication, for instance what queries and synonyms or general engineering terms may be input and understood
- common sense domain or general knowledge about time, sensible heuristics for dealing with unexpected errors, etc
- basic strategies or very general problem-solving strategies or knowledge to aid in system control
- computer language and environment, in common with any other software

These definitions are not hard and fast. For instance, the task domain can be split into

- the concept level that contains the knowledge about the ideas and physical objects which are particular to the problem
- the methods and rules level that gives the particular information on how to deal with the information about concepts, including non-general problem-solving strategies

Finally, a possible model for a knowledge base is given by Laurent (1987) as:

$$\boxed{\text{Knowledge Base} = (\text{ Objects} + \text{Actions} + \text{Halt condition })}$$

where the *Objects* are the defined concepts knowledge base or input and output data, *Actions* are procedures or deductions carried out on Objects and the *Halt condition* is satisfied when the problem has been solved. This model provides the missing link. In general problem-solving it is necessary to provide at least one Halt condition or goal. In RISC, there are likely to be many goals, since there will be a Halt condition for each stage of analysis, interpretation of results, and scheduling of actions.

The first three in the list of layers of knowledge above, namely the task, interaction and common sense domains, are made explicit in KBS. Thus, the problems of knowledge representation are the problems of representing and handling the information in these layers only. The following describes some knowledge representation schemes.

4.2.1.2 Logic

A basic axiom in logic is that if a fact derived from a particular set of propositions is both true and false, then anything at all can be derived from those propositions. Thus contradictions in the knowledge base and input must identified and removed as soon as they appear: in essence the knowledge base represents a closed world. But there can be no guarantee that all contradictions can be identified for any non-trivial system. Since any interesting engineering problem is definitely non-trivial and not a closed world problem, then formal logic alone will not be useful for an engineering expert system.

The above is an over-simplification: formal logic with its grammar and syntax cannot be used to represent and reason about all the information to be used by a software system (Addis, 1985). Still, knowledge representation based on first order predicate logic, that is, allowing the use of variables, does play a major part wherever formal deduction is required or in reasoning about time and beliefs, both of which are required for common-sense reasoning (Long, 1989; Reichgelt, 1989). First order predicate logic is often used in theorem-proving applications and in parts of the process of understanding language. It is also applied to the function of checking databases and knowledge bases for inherent contradictions. In these applications the number of logical statements is small, only a few hundred facts and tens of hypotheses, as compared to a realistic engineering application.

4.2.1.3 Production Rules

The first generation of expert systems, directly based on Newell's "shrieking daemons" concept, were *production rule systems*, that is systems which used sets of rules as their knowledge, and this formalism was widely adopted for consultation KBSs (Davis et al, 1977a).

Almost any item of knowledge can be written in the form of a *production rule* (Frost, 1987). The basic form of a rule is shown in Figure 4.3:

```
If      condition(s)
then    conclusion(s) or action(s)
```

Figure 4.3 Structure of a production rule

where the conditions may be complex conditions formulated using Boolean operators, that is, AND, OR, NOT, FOR ALL, etc. Usually only one action or condition is applied, but several actions or conditions may be listed. Examples are shown below.

```
If      the current probability of failure of a
        component is greater than 0.5
then    component must be inspected now
```

```
If      the node under consideration is a T-joint
        and its SIF is unknown
then    apply the Newman-Raju parametric equation
```

Figure 4.4 Example production rules

As it is easy to express information as rules, they are also easy to understand. It is also easy to construct a knowledge base containing many rules, since a rule can contain one very simple item of information. Creating a knowledge base containing a large number of rules leads to huge problems in maintenance of the KBS, apart from the problems of actually executing the KBS. Thus, one disadvantage of production rules is that they provide too simple a format to represent concisely the large amount of knowledge typically required to solve a real-world problem. Another disadvantage is more fundamental: the user query

```
User        Why should the component be inspected now?
```

would get the answer

```
Machine     Because its probability of failure is 0.65
```

which is too superficial for the user to understand the basis of the reasoning. Production rules provide a method of representing only superficial information on the domain in question.

In general, rules are used for relatively simple or narrow domains and not considered suitable for the integration of information of many different types. Alternatively they are used to express heuristics for problem-solving strategies, which are sometimes only approximate and not to be held to be always the best solving method.

4.2.1.4 Semantic Nets

As it became apparent to AI researchers that production rules did not always represent relationships efficiently, other formalisms were investigated. In particular, various knowledge representation schemes were developed that are seemingly very different, but are based on *associative nets* or *semantic nets* (Charniak & McDermott, 1985).

Semantic nets make explicit relationships between concepts and objects. For instance the platform Brent A is related to the general concept of fixed offshore platforms by the special relation *is_a*:

```
Brent A is_a fixed offshore platform.
```

Similarly, other relations between concepts and objects can be defined:

```
node G2143 is_part_of Brent A

chord 234 is_part_of node G2143

chord 234 is_made_of Material BSD2125

node G2143 is_a K_joint

K_joint is_a_kind_of tubular joint
```

Diagrammatically, these relations form a net-like structure, which could be extended to provide all the information required (Figure 4.5).

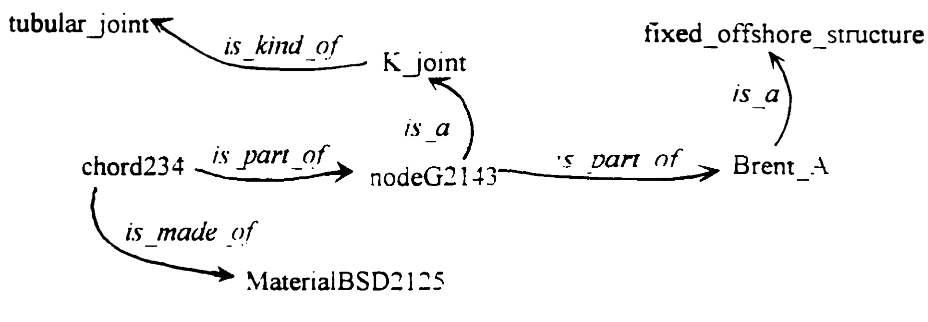


Figure 4.5 Example semantic net

To implement a net as above would entail setting down each relation as a separate item of information. To be able to reason with such a knowledge structure, requires an understanding of the special relations *is_a, a_kind_of* and *is_part_of* to provide more information about any one object in the dynamic knowledge base. The *is_a* and *a_kind_of* relations describe specialisation, which is described further in Section 4.2.1.5, and *is_part_of* transfers information about one concept to another.

Semantic nets are a most appropriate and powerful way of describing relationships between concepts and objects, that is, each connection between each item in the diagram. Unfortunately, there is no immediate way of making the meaning of each relation explicit or independent from the control mechanism. This problem poses a restriction on the development of systems using pure semantic nets as the allowed relations have to be predefined.

4.2.1.5 Schemas: Scripts, Frames and Objects

The format which best allows the construction of large knowledge bases that are not always complete is that of *schemas*, *frames*, *scripts* or *objects*. The names may be different, but the general principle holds: schemas are best described as a set of *slots* and they provide a framework for recognising and storing concepts, whether physical entities, abstract ideas or possible scenarios. The meaning or interpretation of these slots will vary implicitly according to the type of representation scheme and explicitly according to facets associated with each slot (Minsky, 1972). Each slot is in effect a relation, as described for semantic nets. Thus a schema has the advantages of semantic nets, and also ensures that the knowledge is structured, that is, all information about a particular item is held in one place.

Scripts are used to identify situations or to follow a set of actions (Schank, 1977). A script might hold the set of events one may follow in scheduling the inspections for a structure. So with scripts the order of the slots is important. For instance, analysing the structure in the light of past inspection results must of course be carried out before interpreting the results, and this in turn must be carried out before ranking nodes in order of unreliability, and so on. Scripts could be also used to describe the procedure used to carry out a series of reliability analyses.

In contrast to scripts, frames are generally used to describe classes of objects and instances of these classes from a particular view-point (Bobrow & Winograd, 1978). One description of a frame is that it is a nested list of slots with associated lists of *facets* which themselves have associated *values*. Slots are properties; facets are aspects of the property and thus give further meaning to the slot. Facets can be used to indicate any restrictions on the expected value for the slot. Some possible facets are *value-type* to indicate whether the value of the attribute is numerical, text, or even a procedure, *default* to allow a default value for the slot, and *monitors* that are procedures carried out when the slot is accessed.

A simplified example is the class of *fixed offshore platform*, which have many attributes in common such as *operator*, *number of nodes*, *materials*, and *current IRM schedule*. As a jacket-type platform would be expected to have more than 100 nodes, but less than 2000 say, thus the *number of nodes* attribute may have an associated facet *in range (100,2000)* say, in addition to a facet indicating that this attribute is numeric. An *if-needed* monitor type facet associated to the *current*

IRM schedule slot may trigger the RISC System to provide a schedule if one does not already exist. An instance of the class would be *Brent A* with its own values for each attribute such as *number of nodes* = 2547.

Objects are very similar in concept to frames and are now part of the main-stream of computer science where lower-level and industry standard programming languages have been developed to include the concept of encapsulating information into distinct objects. This is referred to as *object-oriented programming* (OOP). OOP has recently gained popularity in computing due to the convenience of class object abstraction as a representational tool. and the power of message passing as an information processing mechanism. Many languages are described as being object-oriented programming languages (OOPL), but there is no agreement on the fundamental principles of OOP just as there is no agreement on the basic principles of schema. Following Stefik and Bobrow (1984), Tello (1989), and Zaniolo (1984 and 1985), some of the most commonly understood and basic concepts or ingredients of an OOPL are

- Objects are complex data structures which combine properties of procedures and data. The structural object tubular joint, for instance, may be represented as a collection of related data elements with associated analysis methods.
- Message passing refers to requesting an object to execute a procedure stored with the object (known as a *method*) and hence interpreted by the object. For example, a message may be passed to the structural object "tubular joint" to obtain its SIF value: the procedure to obtain the SIF is local to the object and so would know which parametric equations were applicable and what values to input into these.
- Specialisation refers to a feature where one object is a more specialised example or subclass of another. The tubular joint object may contain a default SIF calculation procedure, but the more specialised small T-joint may have a particular SIF equation that is best used. This is a very similar concept to that of inheritance which is explained below.

OOP is useful for the unification and simplification of the description of entities and their protocols. There is also an inheritance network set up, whereby an object which is a specialisation of a first object inherits all the properties and methods associated with this first object. An action is carried out by passing a message to an object requesting one of its methods to be executed.

In an OOPL, most, if not all, of the following or equivalent infix operators will be found: *equals* that associates an attribute and value to an object; *sub* and *is_a*, both of which define the inheritance network; *with* that associates methods to objects; and *:* used to indicate a message. As examples, it could be stated that "J100 *equals* the component with the lowest reliability" which would provide more information or identify a particular property of interest more explicitly. "J100 *is_a* T-joint" and "T-joint is a *sub*-class of tubular joints" provide a framework for inheritance of attributes of interest and methods, and default values or restrictions on values for some of the attributes. Stating "tubular joint *with* SIF calculation procedure" associates a method with the class of tubular joints, while "run J100:SIF calculation procedure" will execute the appropriate program for J100.

- **Inheritance and Cross-referencing**

Relationships between schemas can be represented by nesting schemas within each other. One schema can be a parent to another schema and this reduces information stored, as it allows *inheritance* of properties and values. Where the information required within one schema is described in much more detail in another schema, then *cross-referencing* between these schemas is carried out.

As an example of inheritance related to tubular joints, then the frame that describes an X-joint would contain a slot or attribute "superclass" with the value "tubular joint". An X-joint would then take those properties and values represented in the tubular joint frame, unless overridden by the X-joint frame. As an example of cross-referencing, consider a frame representing tubular joints. The material property of an instance of an X-joint might be described as BS4360-50D within the X-joint frame. The detailed information on the material properties of BS4360-50D required for the analysis would be obtained from the BS4360-50D frame.

- **Stereotypes**

A most important aspect of schemas is that default values for parameters may be represented. Default values represent stereotypical objects or situations, which can be used in several ways. At the simplest level the default values can provide aid to the user of a system by suggesting values to be input and hence a stereotype allows an expert to help the system when there is a gap in the user's knowledge. At a more complex level, stereotypes can be used to match actual situations or objects against expected situations and objects as part of the deduction process, or to provide the next step to be carried out, in the case of scripts, or more data for frames. Default reasoning is of great value in real-world situations since our knowledge of a real problem is often incomplete (Reiter, 1978). Only by providing operators indicating defaults, such as may be used in frames, can default reasoning be carried out.

- **Summarising Schema**

Schemas provide a powerful way of modularising the knowledge required to solve a complex problem. By predefining specialised slots, documentation can easily be stored with the information to which it refers. Inheritance and cross-referencing provides methods of re-using information and hence reducing information storage. The main disadvantage is that there are no well defined semantics for schema.

4.2.2 Reasoning and Induction Methods

There are many basic reasoning and control algorithms which may be implemented in a KBS. By "basic" it is meant that these algorithms tackle the problems common to all KBS and hence are found in nearly all intelligent systems. They are:

- Searching strategies are algorithms for controlling the order in which the solution space is searched when investigating a hypothesis or attempting to reach a goal state.

- Pattern-matching methods are procedures that take a fact or hypothesis in the dynamic knowledge base and matches this against the static knowledge base. In effect two patterns are compared and if they *match* in the sense defined by the KBS itself, this indicates that the item of static knowledge can be used to add more information, either facts or hypotheses, to the dynamic knowledge base.

Searching techniques are of great interest in planning applications and hence will be discussed in some detail in Section 4.3. Pattern matching, which is to a certain extent akin to pattern recognition, is one of the most common ways of reasoning with frames, in that by matching an instance to a class, extra information in the form of default values can be provided (Minsky, 1972). First, the global reasoning or control strategies are discussed.

4.2.2.1 <u>Backward and Forward Reasoning in Production Rule Systems</u>

A production system consists of a database or dynamic knowledge and rule base, containing production rules, and a rule interpreter or inference engine (Frost, 1987). The classical reasoning method originally proposed by Post in 1943 is *data-driven* or *forward-reasoning* where the condition part of a rule is matched against the available data. If the conditions for a rule are satisfied then the conclusion can be added to the database. In *goal-driven* or *backward reasoning*, a goal is set and the conclusion parts of the rule-base are matched against this to eventually identify what data is required to attain the goal.

The advantages of a production system are that rules are modular, which eases maintenance but only if the rules are independent, and a natural form of knowledge representation; explanation facilities can be added; and uncertainty in the information can be relatively easily represented by the use of probability values associated with each rule. Some of these advantages are shared by the other representation schemes. The disadvantages of production systems include

- to be certain that a rule will be used only in a very particular circumstance ALL contextual information must be included in the conditions of each rule
- dependencies are difficult to spot in large databases
- the choice of rules and search strategy is not trivial
- most production rule systems cannot handle generalised rules, such as

```
For all nodes,
if node has not been inspected in the past six years
then include in next inspection schedule
```

In summary, a production rule system is only appropriate for narrow domains and not for the integration of information of many different types.

4.2.2.2 Models and Model-based Reasoning

To reason about complex systems, such as the degradation with time of a fixed platform, requires the use of *deep knowledge.* By deep knowledge is meant knowledge which contains the basic principles required to model a system (Cohn, 1985; Fulton & Pepe, 1990). Production rules usually contain only *shallow knowledge*, that is, rules only associate an input fact with an output fact or action, but often this does not include the knowledge as to why the association exists. Thus an explanation for any reasoning based on rules is necessarily limited. The use of deep knowledge, on the other hand, allows greater flexibility in reasoning and better explanation facilities. In order to distinguish between KBSs which make use of simple heuristics to solve a problem and KBSs which model the behaviour of the system in question, to some extent, the term *model-based system* was coined (Fulton & Pepe, 1990). Reasoning in these systems can be carried out by making use of causal models which represent the structure and function of the domain objects and processes.

The problem remains to choose and define the structures used to model the domain. There is a need to have standard conceptualisations or data structures for this purpose. The use of concept discovery systems that carry out the process of extracting information automatically are described by Davis and Lenat (1982). Such systems would require a formal language which

- ensures *fidelity,* that is, that the model does represent the intentions of the knowledge engineer and for this the semantics need to be clearly defined
- provides a many-sorted logic to allow express taxonomic knowledge, such as *is_a* and other relationships between entities
- enables consistent knowledge bases to be created, and formal methods for this include the

use of *truth maintenance systems*, the details of which are beyond the scope of this thesis but are given by Stanojevic et al (1994)

- minimises unnecessary inference chaining, that is provides methods reliant on *meta-control* information (Maes, 1988) to check the likelihood that rules or procedures leading away from the goal can being triggered and to detect circularities in the reasoning process

Standards require being formulated for such languages. Most of these issues are advanced topics for research and discussion and hence are no longer discussed here.

4.2.2.3 The Blackboard Model

A blackboard system is a highly structured opportunistic problem solving system that prescribes the organisation of domain knowledge and data, but not the control mechanisms (Nii, 1986a and 1986b). A system can be called a blackboard system whatever its control algorithms. Thus the term *blackboard* describes a model or concept and does not specify the architecture of a system. The basic elements of a blackboard system are:

- knowledge sources that are modules encapsulating the problem-solving knowledge
- blackboard data structure or a global database that stores the dynamic knowledge of the current problem solving session
- control, although not prescribed, is of course required to control how the knowledge sources can respond to changes in the blackboard

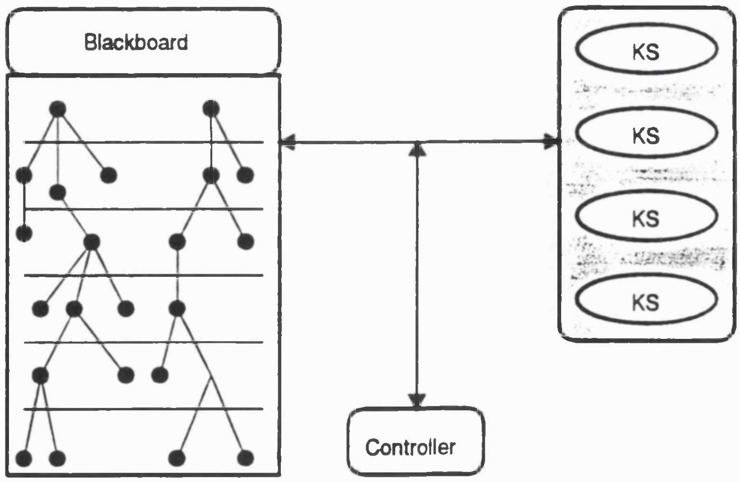


Figure 4.6 A simple blackboard architecture

Historically, the concept of a blackboard system was outlined by Newell in 1962 (described earlier as "a classroom with shrieking daemons") and became the basis for the first production rule systems, where the knowledge sources (or daemons) are production rules which responded to changes in the dynamic database of facts (Winston, 1977).

Nowadays, the term is most often reserved for an opportunistic KBS with a partitioned blackboard, each partition usually representing a *level of abstraction.* As an example of this, consider the example used to illustrate the use of one of the earliest and classic blackboard systems, HEARSAY, that is image processing of photographs to detect and identify koala bears (Nii, 1986a). The lowest level of abstraction may be the raw vision data, the next level may be processed vision data, such as blobs and their parameters, then collections of blobs to make features such as circles or oblongs. Characterised features such as eyes and muzzles or heads, legs, torsos and paws, follow and the final level of abstraction would be the recognised bears. Thus, the higher levels of abstraction usually contain hypotheses inferred from the lower levels.

The blackboard model is most suitable for problem areas such as speech and image recognition or complex signal analysis, where the opportunistic behaviour is required in an attempt to shorten computation time and where the concept of different levels of abstraction fits in well with the language used to solve the problem.

A knowledge representation scheme is not prescribed just as the model does not prescribe the control required to implement a blackboard-based system. A knowledge source can be a production rule, or a schema with some implicit reasoning method associated with it, or a procedural program. As a model it lends itself very well to distributed processing, although how it is to be implemented for parallel processing is another research area in itself (Corkill, 1988).

4.2.3 Modelling Uncertainty

In an engineering domain such as inspection scheduling of offshore structures, much of the knowledge stored will include elements which are uncertain. This arises from or because of

- random variables, such as wave loads, which can be tackled by using well-established methods from statistics and probability
- model uncertainties due the imprecision inherent in a mathematical model of a physical system
- numerical errors which occur when using quantitative methods to evaluate mathematical models
- verbal or qualitative descriptions, which more often than not include an element of *fuzziness* or imprecision

- non-monotonic logic, that is the use of reasoning which changes the conclusions arrived at according to the order in which deductions are made
- imprecision of knowledge, such as production rules representing experiential knowledge which are not always applicable, or default values employed in frames
- granularity of knowledge, where data is given to a certain degree of accuracy in one context, but which in another context may be required to a higher level of accuracy
- incompleteness of knowledge, where a belief can be stated about the truth of some statement, allowing for perhaps ignorance of other information
- human errors occurring during the construction of a knowledge base

The first three cases are dealt with by mathematical algorithms and have already been discussed and considered in Chapter 3. The last is a question of ensuring that good procedures are followed in the development of a knowledge base system and of providing tools with the knowledge base editor to allow the developer to check for errors. The remainder, however, are problems of knowledge representation. Several ways have been developed to represent uncertain knowledge for an expert system or KBS. The two main methods are Bayesian methods and fuzzy logic.

4.2.3.1 Bayesian Methods

In AI systems, Bayes' theorem is often applied in an approximate manner (Saffiotti, 1987). Formulae are derived from Bayes' relation between $P(H_1|E)$, the probability of the hypothesis H_1 being true given evidence E, and the known prior probabilities $P(H_1)$, $P(H_2)$, ... $P(H_n)$ where H_1, H_2, to H_n are possible exclusive and exhaustive hypotheses, and $P(E|H_i)$, the probability of the evidence given the H_i hypothesis occurring (see Chapter 3 for a description of Bayes' theorem applied to decision theory).

One of the earliest attempts to employ Bayes' theorem to model the uncertainties in reasoning was in the MYCIN system for medical diagnosis and in the geological system Prospector (Buchanan & Shortliffe, 1984; Duda et al, 1979). In these systems, conclusions could be qualified with certainty factors, but these are not Bayesian probabilities in the sense that exclusivity of all hypotheses was not required. In spite of this, the calculation of the new uncertainties was based on Bayes' rule. As this is an ad hoc method and only loosely based on Bayes' theorem, the resulting numbers were not well-defined and hence difficult to interpret. This makes the basic method questionable. Other applications of Bayes' rule, in expert systems shells such as SAGE and Micro-Expert are also ad hoc methods (Mamdani et al, 1985). Even when probabilities for all possible events can be found, there are practical problems. To apply Bayes' rule requires being able to state and store all the probabilities (or certainty and belief factors) for each hypothesis, for each rule and for the joint probabilities. This requires a huge amount of data. Thus the Bayes' approach has not been particularly successful and because of this, other methods have been considered which do not attempt a quantitative approach, and instead make use of linguistic terms to indicate uncertainty. These will not be covered here.

An extension of Bayes' rule, which is theoretically more sound than the ad hoc methods outlined above, is the Dempster-Shafer Rule, which makes use of measures of belief in causal relations or the rules relating knowledge, and uncertainty in the evidence being provided (Chatalic et al, 1987). This method can be applied to *fuzzy sets* and used to describe *fuzzy rules*, hence fuzzy logic is next described.

4.2.3.2 Fuzzy Logic

An example was given earlier of a piece of knowledge representing qualitative information about a physical system:

`A brace with a` *`heavy`* `load will buckle.`

Consider a system which contains the above knowledge and information is given to it that a load of 10000N has been applied to a brace. The system will require some understanding of what is meant by a *heavy load* to decide whether or not 10000N will cause buckling. In this context the term *heavy* in the above example is an imprecise or *fuzzy* term. One method of dealing with such a term is to apply *fuzzy logic*, which originated from Lofteh Zadeh in 1965 (Zadeh, 1974 and 1988). It is applied in control theory and now also in business domains, as well as in KBS systems (Griffiths et al, 1987). Its purpose is to enable reasoning based on linguistic variables to be carried out in a systematic way.

It has been argued that the best way to approach the problem of linguistic imprecision is to do away with the linguistic terms altogether and to provide instead a range of possible values or a statistical distribution to model the variable (Kosko, 1992). An argument against this approach is that by requiring a precise numerical description, the representation of the knowledge becomes less comprehensible, that is, if an expert reasons qualitatively, then forcing the use of quantitative terms makes the expression of the knowledge more difficult.

Fuzzy logic provides a basis for the translation of qualitative terms to quantitative values and vice versa, by associating probability functions with terms such as *heavy*, to provide an estimate for a *heavy load*. To apply fuzzy logic requires the definition of *fuzzy sets*. This is done by assigning a probability value to the fuzzy term for associated numerical values. As an example, the fuzzy set "Normal Loads for this particular brace" may be defined as a graph representing membership of actual load values to the set.

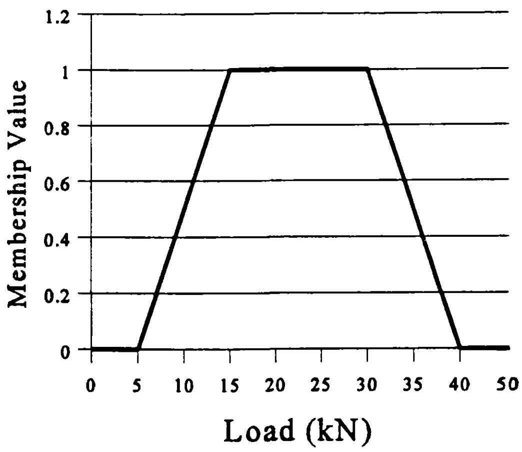


Figure 4.7 Membership values for the fuzzy set "Normal Loads for a brace"

The graph in Figure 4.7 has a typical shape for a membership graph. From it, it can be deduced that a load of 20000N is usual for a brace, whilst a load of 10000N could be considered to be perhaps a little low. Other fuzzy sets can be considered which are related to this set. For instance the set of 'Heavy Loads' and of 'Light Loads' may be defined as in Figure 4.8(a) and (b).

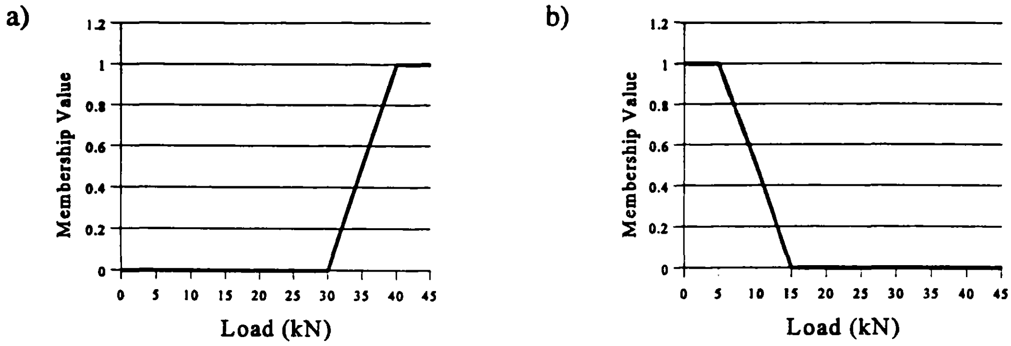


Figure 4.8 Fuzzy sets (a) 'Heavy Loads' and (b) 'Light Loads'

From these, the modified set of 'VERY Heavy Loads' can be defined, where the line is shifted in some direction, 'NOT Low Loads', 'NOT Low Loads AND Heavy Loads' and so on. This combination of various fuzzy sets with other sets and modifiers is carried out by a series of *fuzzy rules*. Suppose that F represents a fuzzy term with membership function p(x), that is, for an object with attribute A set to value x, then A is F with membership value p(x). For a particular brace B, with applied load L=32000N, and membership functions for Normal, Heavy and Light loads as shown in the above figures, L is Heavy with membership value 0.2, Normal with membership value 0.8 and Light with membership value 0. Then example fuzzy rules may be

1. For negation, that is, NOT F, where F is a fuzzy term with membership function p(x), a common procedure is to take (1 - p(x)) as the membership values for the new set. This

transformation does not always lead to consistent results. For example, if NOT Light Load is interpreted as Heavy Load, then the membership function for Heavy Load should be $(1-p_L(x))$, where $p_L(x)$ is the membership function for Light Load, which is inconsistent with the graphs shown.

2. For the conjunction of two fuzzy terms F1, F2, that is F1 AND F2, it is usual to take the minimum values of the combined graphs. So a membership function for a Heavy AND Normal Load would start at 0 for a load of 30000N rise steadily to a value of 0.5 at 35000N and descend to 0 at 40000N.

3. For the disjunction of two fuzzy terms, F1 OR F2, the procedure is to consider the sum of the two graphs, and if any values are greater than 1, the sum is truncated so that the maximum value is 1. For a Heavy OR Normal Load, membership values start at 0 for a load of 5000N, rise to a value of 1 at 15000N and stay at 1 for loads greater than 15000N.

4. For the use of modifiers, e.g, VERY, QUITE, the graph may be shifted either up or down, above or below original line.

Note that a new fuzzy term made up of two or more combinations of the original fuzzy terms may not reach 1 on its graph, e.g. 'Heavy Load AND Light Load' is everywhere 0. If a single value is required, perhaps to be given to the user or to be used as a default value in a calculation, a method of 'de-fuzzy-fying' is required. Several strategies to do this are reasonable, such as "take the value with the highest membership value", or "calculate the central moments of area under the graph".

A comparison of fuzzy logic with classical logic is given in Table 4.1.

Table 4.1 Comparison of fuzzy logic with classical logic

	Classical Logic	**Fuzzy Logic**
1	Values of propositions can be TRUE or FALSE only (Law of Excluded Middle)	A proposition can be partially true
2	Quantifiers are ALL or SOME (or There exists at least one)	Many quantifiers eg: MOST, ABOUT, FEW, SOME
3	Classical inference allows only reasoning based on absolutes, e.g., J101 is a T_joint AND T_joints have one brace) IMPLIES J101 has one brace	Dispositional inference allows reasoning with probables, e.g., MOST tubular joints have cracks at installation AND J101 is a tubular joint IMPLIES J101 probably has cracks at installation

A possible disadvantage of fuzzy logic is the requirement to define the set membership function, and how this is done can be very subjective. Fortunately, it has been shown that fuzzy systems are not sensitive to the exact form of the fuzzy sets. The technique is a robust way of allowing reasoning based on qualitative terms as might be employed by human experts.

4.2.4 Knowledge Modelling for RISC

It is expected that for an engineering domain, such as IRM scheduling, a model-based system may have analysis procedures which model the physical behaviour to various levels of detail. For example, a chord with a particular load may be modelled at a macro-level by considering the chord as a cantilever bending under a particular load, and to a micro-level by involving considerations of the material deformation at molecular level. The use of qualitative reasoning makes use of knowledge representing qualitative information about a physical system, such as the previous example:

```
A brace with a heavy load will buckle.
```

There are various possible ways of dealing with such information, but in order to be able to integrate this type of information with quantitative knowledge requires the ability to transfer smoothly from one mode to another. To do this would need a method of translating from qualitative terms to quantitative measures, and vice versa, such as fuzzy logic. There is a requirement that the RISC System act purely as a consultation system, where the maintenance expert makes all final decisions. Thus, any uncertainties on what values may be used at different points would be presented to the user. Any inherent uncertainties in the knowledge of the problem would be expected to be incorporated into the structural reliability analysis. Thus the use of an AI based procedure for dealing with uncertainty may be wholly inappropriate for the RISC methodology.

The formalism which appears to be of value to the RISC System application is that of schemas for general data storage. Since the analysis process requires a great deal of data to be collected, the slot-and-filler representation provides a highly structured way of storing and managing data. The use of methods or monitors, associated with each class, allows the definition of procedures to calculate missing data values, such as stress concentration factors, which will be particular to each structure type and perhaps even to an instance of a tubular joint. In addition, there will be some fairly unstructured information on heuristics which will constrain as well as aid the scheduling process. It is expected that these rules will be simple rules for data-driven reasoning, which can be structured into small rule sets and brought into play at very specific points of the RISC procedure. For instance, when choosing possible points in time for inspection, a simple rule will check for any data about the joint which indicate restrictions on the period for inspection.

Most of the RISC procedure is quite clearly prescriptive in that a particular set of steps is to be followed. Thus, scripts may be implemented to control the analysis process. If necessary, specialised schemas can be set up for the representation of the more complex heuristics used to aid scheduling as well as overall control of the RISC system. As one of the main tasks of the RISC System is to carry out scheduling, the next section considers the use of AI in scheduling and planning applications.

4.3 AI FOR SCHEDULING AND PLANNING

Scheduling and planning are decision processes which involve selecting the most appropriate sequence or set of actions to be carried in the future given an initial state. Yang (1997) distinguishes carefully between the two terms. Planning is the production of a description of how a sequence of events are interrelated, which would include the definition of the constraints on the sequence of the events. The aim of scheduling, on the other hand, is to select an optimal set of events from a plan which satisfies certain conditions or constraints.

Scheduling and planning are real-world problems to which mathematical techniques borrowed from operational research and management science have been applied, such as dynamic programming for assembly scheduling employing robots (Jiang et al, 1997). Although these methods are often effective in certain situations, for the general scheduling problem they can fail (Berry, 1992b). AI has been applied to this area in an attempt to take into account all the problems that arise in scheduling which cannot be taken into account by the use of mathematical techniques. This section considers some of the issues and searching techniques which may be relevant to the RISC System. It also proposes a planning and scheduling algorithm for the RISC System.

4.3.1 The Scheduling and Planning Problem

Scheduling involves allocating resources to perform a series of tasks over a period of time. The decision problem in scheduling is to how to assign actions to points in time, to achieve the goal of ensuring that all the actions are carried out within some stated objectives. This may require reasoning about time, resources and other constraints. Le Pape (1993) defines two basic categories of planning or, in this context, scheduling problems:

- pure resource allocation, in which the demand for each resource is known beforehand and the problem is to ensure that resources are allocated such that availability equals or exceeds the demand

- pure scheduling, in which the capacity of each resource is known beforehand and the problem here is to schedule activities such that the demand on each resource is less than or equal to the availability

Realistic scheduling and planning problems are a combination of the two categories.

Planning problems involve ordering the sequence of activities. This requires identifying precedences, that is, any relationships between activities which dictate the order of execution. For instance, if a detailed IRM schedule were required, it would be necessary to realise that inspection of a joint must be carried out before a repair, thus inspection has precedence over repair. For activities with no direct relationship, it may be possible to run these in parallel. So, if two divers are employed in inspecting joints, then two inspections may be carried out at the same time. Plans in which some activities are carried out in parallel are termed *non-linear plans* since the plans can be modelled as a network, rather than as a linear sequence, or activities (Navinchandra et al, 1988).

The RISC planning and scheduling problem, on the other hand, does not require a detailed schedule that shows the order of the activities in each scheduling period. Thus, it is not necessary to consider precedences between inspection activity and only some parallelism may be considered for the possibility of more than one diver being employed. The decision to enable more than one inspection to occur at the same time is in fact not the operator's, but the diving subcontractors and hence has little bearing on the operator's problem. Thus, the general problem of network planning will not be considered here.

The researchers in the AI field are interested in planning on two levels. On the one hand the planning problem is one which occurs in many situations. The human is a planning animal that will formulate detailed programs of actions to achieve some goal. In addition, AI programs require an element of planning to achieve objectives that lead to the goal of solving a particular problem. In both cases, there is the common aspect of intelligent agents trying to accomplish goals.

Planning can be considered as a searching problem, since the planning problem can be stated as requiring to find a solution out of all the possible actions, which achieves the required goals (Georgeff, 1987; Tate et al, 1985). Planning can also be carried out by applying case-based reasoning. If a past plan succeeded then the chances are that a similar plan would succeed again , as long as the objectives remain the same and there have been only a few changes to the possible actions and constraints (Tate et al, 1985). This reflects the reasoning which is carried out in most industries when old plans are re-used, but if these no longer achieve the goal, then they are modified by trial-and-error until a new schedule is produced (Duchessi & O'Keefe, 1990). In this section, methods which are based on searching techniques and on dealing with constraints will be considered in some detail.

4.3.2 Tree-Searching Techniques

One way of searching through a range of possible combinations of decisions is to consider the solution space to be a tree of possibilities otherwise known as an *OR tree* (Tate et al, 1985; Bolc & Cytowski, 1992). A node of an OR tree represents one event and the arcs from each node represent alternative events, only one of which can occur. In the tree of possibilities, each node is a decision and from each node, there is a another set of possible options from which to choose (cf. the decision theory tree described in Chapter 3 for the choice of best inspection). A general and incomplete search tree is shown in Figure 4.9.

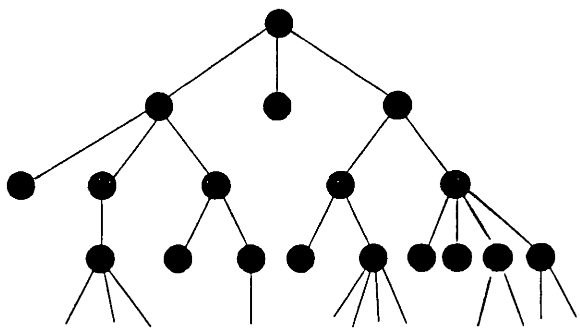


Figure 4.9 A general search tree

As shown here, the root usually represents the decision to find a goal state. The leaves or end nodes usually represent a possible solution, or if not a solution, a point at which no further options exist. The route or path from the node to leaf node represents the order in which decisions are made. In the example tree, some paths appear to continue further and it is possible to have an infinite tree where some paths never terminate. Associated with each node is a cost or value. The basic problem now is to find a leaf node which represents the best solution or at least a good solution. The factors to be considered are the total costs for each route down the tree, and the computational effort required to generate each new set of decisions and to calculate the costs.

The issues to be considered are how to create such a tree and the best way to search through it. The first is particular to the domain of interest. The second issue is common to all tree-searching and some techniques are described here.

The ideal situation would be to use the tree structure to organise the solution space to carry out a systematic evaluation of the costs for all possible solutions, which is in effect an exhaustive search through the tree. It is only necessary to consider a very simple binary tree with a depth of three layers, as in Figure 4.10, to realise that this is not without problems. The number of possible solutions is 2^3. An exhaustive search and evaluation of all the possible solutions for this simple tree requires at least $(2+2^2+2^3)$ computations to generate each possible path.

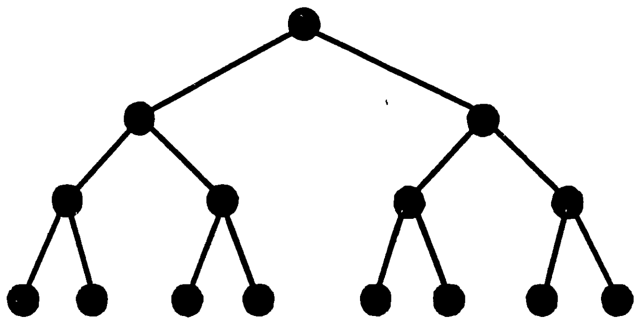


Figure 4.10 A simple binary tree with 3 layers

For a tree with more than two options at each node and more than three layers, the number of computations increases exponentially and hence very rapidly. This is a classic example of what is called in computer science and mathematics an *NP-complete* or *NP-hard* problem, that is, it is computationally intractable since there is no algorithm which guarantees to solve the problem in polynomial time (Garey & Johnson, 1979; Atabaksh, 1991; Pearl, 1984).

Some classic brute force searching techniques are now described. These techniques do not guarantee finding a optimal solution, or even any solution in a specific amount of time for a general problem. By careful consideration of the type of tree and problem in hand, it is usually possible to apply one of these techniques fairly successfully. By exploiting the restrictions on the domain, it may be possible to find an algorithm which leads to a solution in polynomial time.

4.3.2.1 Depth and Breadth First Methods

Depth first and breadth first searches are the most fundamental tree searching algorithms. In the depth-first algorithm, the tree is searched in such a way that the nodes further down the tree take priority.

Consider the tree in Figure 4.11 and note that the form of the tree is not known beforehand, that is, nodes are identified only as the search proceeds. The figure represents a search which is still in the process of finding a suitable solution. The numbered nodes have been identified in the order of the numbers and four possible solutions, 1-2-3, 1-2-4-5-6, 1-2-4-7 and 1-2-8 have been evaluated in that order. If any solution was acceptable then the search would have stopped at 1-2-3. The search shown in this figure implies that there is a requirement for some criteria be met, such as the cost of the solution be less than some value, that was not satisfied by the solution given by 1-2-3.

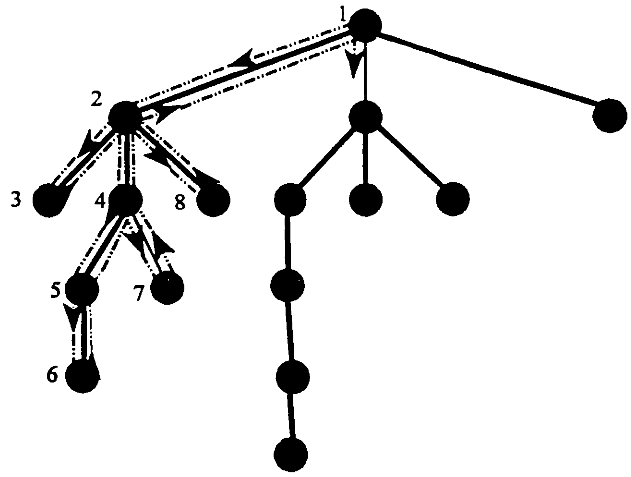


Figure 4.11 Depth first search

Where the tree is not infinite, that is all paths terminate, as in this example, then a depth first search is a feasible method as at least one solution will be eventually found. This example tree does have one rather long path yet to be traversed. If the solution were to be a path lying on the right hand side of the long path, then this procedure would spend much computational effort on one path only. If there is an infinite path, then depth-first may never come to a solution. This last problem is overcome by setting a maximum number of nodes in any one branch or path which may be searched and thus forcing the search to jump to the next branch.

Breadth-first searches prioritise searching through nodes closer to the root. For the same tree, and again remembering that the precise form of the tree is unknown until the identification of the node has taken place, a breadth-first search could be illustrated as in Figure 4.12.

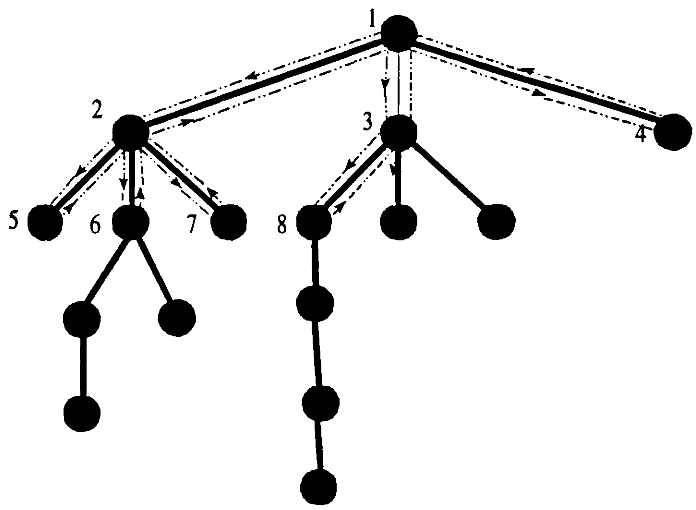


Figure 4.12 Breadth-first search

In this search, three possible solutions have been identified so far in the order 1-4, 1-2-5 and 1-2-7. Thus short paths are very rapidly identified and evaluated. This example demonstrates one advantage of this method, that is that if at least one path is non-terminating, the search for at least one solution will be successful. Yet it has been assumed so far that at each node the number of options are at least finite and ideally fairly small.

For a tree which may have many options at any one node, breadth-first may spend too much time in partial solutions and if the number of options is infinite, breadth-first does not terminate. This last problem is relatively easily overcome in that by setting a maximum number of nodes at any one level of the tree that may be searched, it is possible to force the search further down a path. Another disadvantage, although not as serious, is that for a tree with paths which are all approximately equal in length, this method will spend much time on the evaluation of partial solutions. In contrast, a depth-first search will reach one solution very quickly and very efficiently. The anticipated form of the tree then dictates which is the best method to use.

These two methods are brute-force methods in that during the searching procedure no knowledge of the problem state is utilised. For instance, costs of the partial paths are not taken into account to direct the searching procedure. Extensions to these methods are easily derived and are discussed next.

4.3.2.2 Best-First Search and Other Methods

One obvious extension to the above methods is to make use of the partial path cost evaluations at any one point and to search the next node on the least cost path. This is known as *cheapest-first* search. It is usually more interesting to find the best overall path and this requires an estimate of the cost of the remaining path. If the cost of the remainder of the search path can be estimated and the least costly of these selected, then this is *best-first search.*

Several variations exist of the basic cheapest-first and best-first searches. One could consider the very best of all the partial solutions found or to be found, the best path at any one level, which is an extension of breadth-first, or the best in the next level along the path, which is an extension of then depth-first algorithm (Korf, 1987). The basic disadvantage of these methods is that extra computation and storage of the costs for each path is required. This is often acceptable given the advantages of the methods. Other extensions include the A* and B* methods (Thornton & Du Boulay, 1992). The first method is a generalisation of cheapest and best-first, where the sum of the partial costs and estimated future costs for a solution are used. The B* method involves estimating optimistic and pessimistic values for complete path to a solution, based in the partial path costs and knowledge of the likely values of future costs (Berliner, 1979). As more partial paths are evaluated, the range of possible values for the paths changes, any path with an optimistic value below the best pessimistic value is no longer considered. The method corresponds to the operational research branch-and-bound method as described in Chapter 2.

Further heuristic-based and opportunistic methods based on making use of knowledge about the problem state and committing or not committing to certain paths are also described in the literature (Gaschnig, 1981; Pearl, 1984; Tate et al 1985; Fox et al, 1989; Bolc & Cytowski, 1992). In particular, the best-first technique is easily generalised by including heuristics to define the *best* node to consider next or to estimate the costs. Additionally, the simple methods described above can be extended to deal with *AND/OR trees*, where AND nodes are partitions of the predecessor node to which they are connected. Figure 4.13 shows an example AND/OR tree where the AND nodes are indicated by the use of curved lines binding the arcs. For a solution to the problem modelled by this AND/OR tree, a path must be found from node 1 to nodes 2 or 3. If the search leads to node 2 then a solution path must be include all nodes below it, that is nodes 4, 5 and 6. Similarly if the search reaches node 9 then the solution must include nodes 12 and 13.

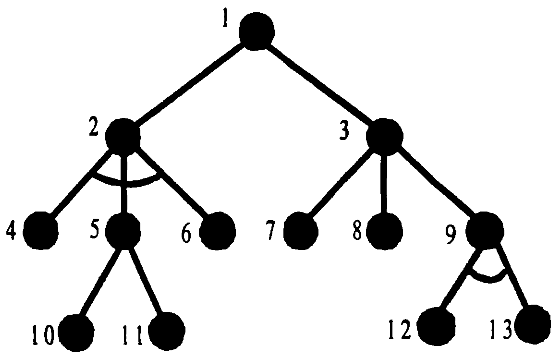


Figure 4.13 An AND/OR tree

More advanced methods include the use of distributed systems to allow several paths to be searched at the same time. These are beyond the scope of this discussion at the moment, but may be of value in future extensions to the RISC work.

4.3.3 Constraints-Based Planning

Scheduling problems can be dealt with by utilising constraints satisfaction schemes which exploit the declarative representation of the constraints on resources in which to carry out the activities to be scheduled (Dean & Wellman, 1991; Yang, 1996). Many researchers have made use of constraints-based ideas in scheduling systems for many different industries (Atabaksh, 1991).

A constraint represents a relationship between plan variables, as a condition which must be satisfied. So, "sub-sea inspections of tubular welded connections can only be carried out during the summer months" is a general constraint representing the notion of a weather window. A more specific constraint may be "all joints to be inspected with eddy current techniques must be inspected in one weather window" representing the operator's concern that the specialist divers trained to use eddy current equipment may be employed for one inspection period only. Constraints are modified during the planning process. As more inspections are planned for a particular weather window, there is less time available for further inspection actions to be assigned to that weather window. Certain constraints are dealt with implicitly by allowing only a certain possible set of values; others are dealt with by employing heuristics or by systematic methods.

One method of dealing with constraints is to carry out a generation, posting and satisfaction cycle (Stefik, 1981; Chandra & Marks, 1986). By *generation,* it is meant that new constraints are formulated based on the current state of the plan. As the plan becomes more refined, then the constraints are also refined. During *propagation*, constraints are associated with a network describing the scheduling process, which here may be a tree of possible IRM actions. Thus, the constraint "sub-sea inspections of tubular welded connections can only be carried out during the summer months" becomes "only one month remains free for inspection tasks in year 3" at the point where only the weather window in year 3 is under consideration and several inspection tasks have been allocated to the weather window. This process ensures that the searching is carried out efficiently and provides a method of reducing the search space at each point of the scheduling tree. The use of constraints to reduce the search space has been applied in many scheduling systems as well as in design systems (Sriram & Maher, 1986; Maher & Gomez, 1996). Finally, after reducing the search space, it is necessary to *satisfy* the constraints.

More systematic methods exist for general constraint satisfaction problems (CSP). Algorithms reviewed by Kumar (1992) and in Bolc and Cytowski (1992) are based on describing a CSP as a set of variables with a finite set of discrete values each variable can take and constraints defined on the variables. The methods are based on describing the CSP as a network or graph where the nodes are the variables and the arcs between pairs of variables represent constraints between them. The methods allow systematic searching and reduction of the graph until no solution or one or more solutions are given. These methods treat every constraint in the same way as well as being restricted to only discrete valued variables.

It can be useful to categorise scheduling constraints to allow different treatments. In the ISIS system, constraints were categorised as hard or soft constraints (Fox & Smith, 1984). Hard constraints represent physical restrictions or legal requirements, which cannot be exceeded and hence these predominate in the creation of an initial schedule. The latter are preferences and can be exceeded if a solution is otherwise difficult to find. The CSP algorithms described by Kumar (1992) target hard constraints. Other researchers have considered ways of modelling soft constraints to allow uncertainties in requirements to be considered. Berry (1992b) describes a constraint satisfaction approach based on using statistical models of the constraints. For instance, models for the start times describe a range of possible times with a preferred start time, and a PCP (Preference Capacity Plan) for a resource gives a likely demand for the resource at different times based on a combination of the constraints, in terms of resource availability, preferences, and requirements.

4.3.4 Applying Tree-Searching and Constraints Satisfaction to RISC

In the RISC System, the planning problem is to decide on the possible inspection actions which may be carried out given certain constraints. The required output is in essence a very simple linear schedule, in that a window of time is allocated as a resource and a list of inspection tasks for a year could be assumed to be a sequence of activities, albeit unordered. The IRM scheduling problem in the RISC procedure is to assign each future inspection task to a node to an inspection period, in such a way that the constraints on resources and the constraints on the inspection tasks themselves are not exceeded.

The domain sizes of the variables are, in theory and at least initially, large, that is there are many possible IRM actions for each joint or node. The resource constraints on the resources are flexible in that an operator can decide to re-allocate resources from one year to another albeit within certain budgetary limits. The more systematic CSP algorithms were not considered in detail, instead a pragmatic approach involving a searching for possible schedules was taken.

As will be described in Chapter 5, the planning problem is solved by applying heuristics specifying hard constraints to narrow down the number of possible maintenance plans per joint and for the structure, then evaluating these by the use of reliability analysis. From this a subset of possible plans based on the analysis results can be selected and this is then the basis for the scheduling problem.

A two-stage procedure is suggested where an initial schedule is then modified to take into account softer constraints. The searching is carried out at this stage of the process. A proposed solution is to combine both constraints satisfaction and tree searching. This can be done by formulating an AND/OR tree representing the scheduling problem for RISC and applying tree-searches to find alternatives systematically whenever it is found that the inclusion of an action means that a constraint is exceeded. The constraints are stated and are easily checked at each stage of the search. Several search methods can be applied to find several possible solutions. Currently the simplest searching algorithms are implemented, that is the depth-first, breadth-first and best-first methods. The solutions found in this manner are complete schedules of inspection plans to be carried out on the structure which do not exceed stated resources and preferred constraints.

4.4 REVIEW OF APPLICATIONS OF KBS

AI concepts have been applied extensively to the whole range of real world engineering problems. A classic application of neural networks is in signal processing, such as in for the interpretation of sensor data during non-destructive testing, where data obtained from eddy current or ultrasonic equipment is processed by a neural network to abstract a classification of the possible defect (Windsor, 1995; Shyamsunder et al, 1995; McNab & Dunlop, 1995). As a contrast to the use of AI for low level data processing, neural networks have also been used in high level design tasks such as prediction of ship container capacity, and have been combined with fuzzy concepts, which enable the use of linguistic variables for input and output, for concrete pile damage diagnosis, concrete mix design and design of industrial roofs (Ray et al, 1996; Rajasekaran et al, 1996). Fuzzy reasoning has also been used in power system control for dealing with linguistic variables describing voltage settings, load demands and so on within a knowledge base system (Laughton et al, 1994). Concepts related to expert systems have been employed more extensively in engineering, than other ideas from AI. Frames have been used in continuous systems simulation modelling a set of first order differential equations, to aid data entry, consistency checks, and the interpretation of results (Nolan & McCarthy, 1986). Case-based reasoning, allowing human-computer cooperative decision-making, has been employed in job-shop schedule optimisation (Miyashita, 1995).

Expert systems have been proposed and developed to aid in the use of statistical analysis, to control the use of software systems and to support the use a collection of analysis programs (Gale, 1987; Newell & Steier, 1993; Apte & Weiss, 1987; Peers, 1997). Object-oriented concepts have been widely adopted, such as in the work on the successor to the IGES CAD data standard, STEP or STandard for the Exchange of Product data, for the representation and storage of engineering data and have also been applied to analysis of NDT sensor data (Bloor & Owen, 1991; Björk, 1991; McNab & Dunlop, 1995). This section concentrates on expert systems and knowledge base systems for general engineering applications or related to reliability-based scheduling and planning. Some typical applications of KBSs include

- diagnosis in medicine and of faults, for example MYCIN and CRIB
- design and configuration systems, such as HI-RISE and XCON
- provision of advice in the use of potentially complex procedures, such as codes of practice and analysis programs, for instance, ELAS and PLAIM
- planning and scheduling for manufacturing, project management, or maintenance, such as ISIS and MOLGEN

The classic, early expert systems are now described, and a review of some recent application of KBSs to general areas in planning and scheduling and in engineering follows.

4.4.1 Early Examples of Expert Systems

Probably one of the first general purpose AI systems that would now be termed an expert system was GPS, or General Purpose System, which was, as mentioned earlier, based on production rules and intended to imitate human reasoning (Newell & Simon, 1963). From this work, the system Soar was developed. Soar was used to develop, in turn, several specialised expert systems (Newell & Steier, 1993). Another early general AI system, described as a "meta-expert system", Teiresias, was developed to provide expertise on methods of knowledge acquisition for the development of expert and knowledge base systems (Davis & Buchanan, 1977b; Davis, 1979). In particular, Teiresias carried out exhaustive depth-first searches to find the best rules from sets of data. This work extended knowledge on the application and limitations of induction and the problems of knowledge representation. The AGE (Attempt to Generalize) system was another well-known meta-KBS and development system which aimed to provide AI scientists with a consistent framework to carry out systematic knowledge engineering (Nii & Aiello, 1979). In other words, AGE provided programmers who were comfortable with AI concepts with guidance and tools to develop and test different representations and reasoning mechanisms.

The most well-known expert system applied to a real-world problem is MYCIN which was a production rule system for medical diagnosis (Shortliffe, 1980). One of the fundamental issues highlighted by this application was the problem of knowledge elicitation, that is how to extract from experts the required knowledge to solve the diagnosis problem correctly and completely, and how to represent knowledge to ensure that it was manageable. The MYCIN system was successful enough to be re-developed as E-MYCIN, or Empty MYCIN, that is MYCIN with no knowledge base, which could then be used to develop other production rule systems. Based on this, many of the early applied expert systems were production rule systems. Dendral was a system which aided in the interpretation of organic chemistry data to identify certain compounds (Buchanan & Feigenbaum, 1978). This was also a rule-based system, but the control strategy adopted was Plan-Generate-Test, which was comprehensible by the experts although it did not reflect the reasoning carried out by the chemists. From this work, further successful systems were developed including INTSUM for interpretation of mass spectrometry data.

An early, commercially successful system was Prospector (Duda et al, 1979). This system was built using E-MYCIN to apply knowledge of geology on the analysis of geological samples to indicate the likelihood and type of ore deposits. The knowledge representation was based on an inference network, with the relations between entities, or rules, being given plausibility values and measures of belief accompanying evidence. It was said to have successfully identified fields positively which were not identified by human experts and thus to have paid for its development very early on. Another early engineering expert system was Computer Retrieval Incidence Bank (CRIB), which was developed to find faults in a computer (Johnson & Keravnou, 1985). CRIB was remarkable for being able to learn from new cases presented to it and was also intended to aid the engineer rather than replace the engineer. This work explicitly restated the role of an expert system as a consultation system, that is, employed by the user as an adviser.

One of the first blackboard systems was Hearsay II, applied to speech understanding (Erman et al, 1980). This system had a partitioned blackboard with the lowest partitions representing sounds and 'chunks' of sounds (segments) and the highest levels representing phrases and the semantic meaning. From this work, HASP was developed for ocean surveillance (Nii, 1986a and b). HASP built up hypotheses of the existence of nearby submarines based on interpreted sonar data using the exactly the same concept of a hierarchical blackboard as Hearsay II.

4.4.2 General Engineering Applications

Following on from the work on MYCIN and CRIB, early KBS for engineering domains were essentially diagnosis systems and much work has continued in this area, for instance in developing models of the structure or system in question and qualitative reasoning methods (Price & Hunt, 1989). Examples include FAUST for diagnosing faults in an electricity supply system (Bramer et al, 1988). FAUST monitors the grid system and can reason, with incomplete data if necessary, by carrying out simulations on an internal model of the grid.

As design is considered to be one of the most complex of problems, many researchers have considered ways aiding designers by the use of AI based software. Design is an inherently open-ended problem, that is, different situations, new environments, new additional information will occur, so it is impossible to have an ideal or even complete knowledge base for design. This is particularly true for the creative or synthetic stage of design which is an indeterminate process (Yagin, 1989). Thus, design systems are required to be highly interactive to allow the user as much freedom to modify or add information to the database being built up on the design model (Rychener, 1985). At deductive, analytical stages, automation can be implemented effectively, and many expert systems exist for detailed design. In the area of structural design, work has been carried out in trying to incorporate structural design codes in an explicit form to knowledge-based design systems, such as that reported by Topping and Kumar for steelwork (1989).

Arockiasamy and Lee (1989) carried out a comprehensive review of structural design expert systems. These included information on design standards. techniques to generate site layouts, knowledge of soil exploration, methods of construction planning including real-time operations to aid the design process for buildings, bridges and frameworks. One structural design system, HI-RISE, makes use of constraints to limit the design solution space (Sriram & Maher, 1986). HI-RISE is intended for the preliminary design of buildings and the output is a set of feasible spatial plans. The idea of constraints has also been employed in mechanical design combined with object oriented databases (Chakrabarti et al, 1992). During embodiment design, that is the stage after conceptual design and before detailed design, constraints placed on the mechanical functions of a physical system are identified and then propagated to generate a description of the physical object. The same idea of modelling with constraints has been proposed for CAD support (Anderl & Mendgen, 1996). Constraints satisfaction was combined with case-based reasoning in CaseCAD, a system for conceptual structural design (Maher & Gomez, 1996). A task which is related to design is that of configuration of a system, that is taking standard parts and combining them to provide a usable system. One well-known system, XCON, was used by Digital Equipment Corp technical salesmen to be able to configure VAX computers for their customers (McDermott, 1981; Buchanan, 1986).

The Archon system is a distributed AI development software platform which is primarily employed in communications and control applications, such as electricity-transportation management and particle-accelerator control at CERN (Jennings et al, 1996). Archon allows the development of problem-solving entities or agents to control and act completely independently if necessary, interacting with other agents if they require more information. Each agent is made up of two layers: the application layer carries out the necessary computation to solve a problem, carry out a process, or whatever; the Archon layer provides the interface between the agent and the rest of the community of agents. This separation of the domain-level knowledge from the interaction between agents allows software developers to use both a bottom-up approach in design by concentrating on the solution of small sub-problems, and a top-down approach in looking at the overall requirements of the complete system.

Another system of interest, but which is very different in its approach, is the StAR risk adviser system which is applied to the toxicological risk assessment, although the methodology is applicable to other types of risk (Hardman & Ayton, 1997). StAR is a decision support system making use of production rules for both supporting and eliminating arguments or conditions. The rules allow multiple hypotheses to be built up based on the evidence. In addition to this system, Comerford and Stone (in Blockley, 1992) list many commercial and research heuristics-based systems for general risk assessment in different engineering industries, such as the evaluation of bridge safety, safety of construction projects and earthquake damage assessment for buildings.

4.4.3 AI based Planning and Scheduling Systems

AI techniques have only relatively recently been used to aid planning and scheduling. The reasons for this may be that mathematical techniques provided many methods to tackle the problem. In addition, scheduling requires much information and computation and it is only recently that the level of computing resources generally available have been sufficient to apply AI based techniques with their heavy requirements for computer memory and rapid processing, to a problem which was also inherently memory hungry.

The early systems in the 80s concentrated in the problem of reasoning with constraints. One such system was the ISIS system, for job shop scheduling, which also provided a method of constraint relaxation (Fox & Smith, 1984). MOLGEN was another early constraints satisfaction planning system for laboratory gene-cloning experiments in molecular genetics (Stefik, 1981). It also approached the planning problem as a hierarchical task, where the system was nearly decomposable into sub-tasks. By providing constraint propagation, the sub-tasks are still linked to each other.

The OPIS system, an opportunistic scheduler based on a blackboard system, works by producing predictive schedules and refining these by reacting to opportunities as well as conflicts in resources requirements (Smith & Ow, 1990). Another scheduling system described by Anandhi et al (1993) is based on a blackboard architecture to allow multi-level reasoning to allow schedules to be updated as required. Extending the idea of distributing scheduling problems across knowledge sources, YAMS, or Yet Another Manufacturing System and DAS, Distributed Asynchronous Scheduler were two of several systems which employed agents for distributed processing of scheduling sub-tasks (Prosser, 1993; Berry, 1993).

Instead of providing a system for a particular industry, a very different approach was taken in creating PECOS (Puget, 1993). This is an object oriented and generic constraint satisfaction software library with which systems for planning vehicle routes and scheduling of crews can be developed.

A KBS for production planning in a light manufacturing company with seasonal variations, is described by Duchessi and O'Keefe (1990). This system builds search trees of the production plans and makes use of best-first searches modified by the use of heuristics to reduce the search space. Another production planning system, PCP, mentioned in Section 4.3.3, considers uncertain constraints by modelling them as statistical distributions and employs constraint relaxation alternatives within a blackboard system (Berry, 1992b). Planning of assembly and other manufacturing processes has also been tackled by a mix of operational research and AI techniques. Jiang et al (1997) explains a two-stage procedure for planning a time optimal assembly sequence for robots in a workcell. An initial sequence is obtained by the use of dynamic programming and then this is optimized to provide a time optimal sequence and taking into account precedence constraints.

Most scheduling systems making use of AI techniques have focussed on constraints management techniques. A few examples exist, however, which make use of a broader range of AI planning techniques. GHOST is a project network generator for the construction industry (Navinchandra et al, 1988). Its knowledge base is made up of "critics" or knowledge sources that work on activities as input to find precedences between activities and from this builds up non-linear schedules incrementally. A practical system is Optimum-AIV, which was developed by the Artificial Intelligence Applications Institute in Edinburgh for the European Space Agency for project management of the assembly, integration and verification of spacecraft (Parrod & Valera, 1993). The system drew on previous work on Nonlin and O-Plan, early experimental project management systems which attempted to take an interdisciplinary approach to planning (Tate, 1977). Optimum-AIV incorporates object-oriented designs for the initial plans, hierarchical planning, consistency checking during plan specification, recording of rationale behind plans, plan repair assistance based on recovery plans as well as constraints satisfaction algorithms.

Other researchers have considered further the overlap between management science and AI. Operational research techniques have been integrated into a KBS to form an intelligent maintenance optimization system, IMOS (Kobbacy et al, 1995). IMOS is a decision support system to aid the formulation of optimal maintenance policies. It provides a general structure that includes a knowledge base of expertise on maintenance strategies, such as planned maintenance and repair policies, and a model base of mathematical analysis techniques to be able to evaluate schedules. The target area is that of large systems of components as might be found in the continuous process industry.

Other noteworthy work includes the development of a job-shop scheduling system, CABINS, which uses case-based reasoning to acquire scheduling knowledge (Miyashita, 1995). CABINS uses a search tree structure to carry out a generate-test-debug process to find an optimal schedule. Schedules are generated by focussing on one job-shop activity at a time. A generated schedule is then "tested", that is, evaluated against a set of criteria including the user's acceptance of the schedule. If the schedule fails to meet the set criteria, then it is "debugged" which here means it is modified by use of some schedule repair method. The user may also suggest a repair method, and this will be stored together with the user's reasons for the repair, as a set of cases, where a "case" describes an application of a schedule revision decision on a single activity. The stored cases can then be used automatically in future scheduling by CABINS.

4.4.4 Use of AI in the Oil Industry

The oil industry very rapidly grew interested in expert systems during the 80s following the early and very successful Prospector system. These early engineering applications of KBS were based on the production rule system architecture. One production rule based example is that of Gasoil (Guilfoyle, 1986) which provided advice at BP International on many engineering tasks on petrochemical rigs. Gasoil was reported to be the second largest expert system of its time in that it has a knowledge base of 2500 rules, which were extracted by induction on examples provided by experts. Another oil company, Amoco Production Company, provided programs used by well-logging analysts to be incorporated into ELAS (expert log analysis system), which is described as a hybrid system including experiential heuristics with mathematical methods (Apte & Weiss, 1987).

Katsoulakos and Hornsby (1988) reported on various expert systems being developed for marine applications. FOCES is a marine fuel oil characterisation expert system developed to aid ranking of fuel combustion behaviour. The procedure is based on a two-stage characterisation, where the first stage identifies a property area within a fuel map and the second stage establishes detailed correlation measures based on an analytic procedure. A diesel engine expert diagnostic system, DEEDS, contains a database of symptoms and associated engine faults and an engine simulation model. Also developed were expert systems which carry out optimisation of maintenance schedules for hull and machinery of ships. Ship control systems and dynamic positioning expert systems were also considered.

Expert systems have been applied to the use of analysis programs, for example, SACON which advises on the use of a structural analysis package, and from the work carried out on the ELAS system described earlier, a general system or expert system shell was developed for the control of software packages (Apte & Weiss, 1987).

At least one KBS has been developed for the assessment of flaws in welded components (Willoughby & Laures, 1990). The knowledge encapsulated by the system was based on the codes of practice specifying the procedures used to apply fatigue fracture mechanics. Another system, MATISS, provided fatigue failure analysis for structures and included knowledge bases storing mechanical and chemical properties of materials (Weiss, 1986). A fatigue advice system described by McMahon et al, (1994) makes use of hypertext within an expert system and is intended for use during design.

Also relevant to this thesis is the work described by Chen et al (1996) on using AI based searching techniques in evaluating structural system reliability. In this work, the systems models used in the β-unzipping Method for estimating system reliability of structures (described in Chapter 2), are re-formulated as search trees. AI based searching techniques are applied to finding the probability of failure for a system more rapidly than by the basic β-unzipping Method.

Of particular interest to the work described in this thesis are the use of AI techniques in the reliability assessment of offshore structures and for deciding on an inspection strategy. Two systems have been developed in the past which attempt to do this. The RAMINO (Reliability Assessment for Maintenance and Inspection Optimization) system aids in carrying out reliability analysis on offshore oil platforms and on pressure vessels of nuclear power plants (Khong & Lucia, 1990). RAMINO provides three levels of knowledge, the analysis programs, object-rules and an overview super-module. Its distinguishing feature is that it provides an non-prescriptive approach to the assessment of the reliability of a structure, which is unlike the approach taken in RISC where the reliability analysis has been carefully defined. It also aims to consider many different failure modes. The other system is PLAIM (Platform Lifetime Assessment through Analysis Inspection and Maintenance) which was developed using a frame-based Prolog system (Langdon et al, 1989; Ahmad et al, 1991). It is intended to provide assessments of fixed jacket platforms and to provide planning for inspection and maintenance tasks. PLAIM, in common with RAMINO, provides a non-prescriptive approach to the assessment of the reliability. Large rule bases, with more than 1000 rules required for 100 tasks, are employed in PLAIM and this may result in problems in maintenance of the knowledge and in control of the reasoning process. The work carried out in developing these systems has succeeded in demonstrating the need to find more efficient and effective ways of providing reliability assessment for offshore platforms and to incorporate this into rational planning for inspection, repair and maintenance of the platform.

A more recent and more general decision support system for safety management developed for the offshore industry is ARMS (Advanced Risk Management System) (Besse et al, 1992). ARMS is an intelligent risk assessment tool which makes use of heuristics and decision theory to extract from the user the required information with which to build influence diagrams. These are analysed making use of a probabilistic approach to model random data and uncertainty reasoning to model uncertain cause and effect relationships. The system is intended to help operators carry out "what-if" analyses to identify areas of concern for any type of offshore structure.

4.5 CONCLUSIONS

As for any other form of model, a computer model can only approximate the real system that it is modelling. In knowledge base systems, the accuracy of this model is in part influenced by the knowledge representation scheme. To summarise, the most promising representation scheme for RISC is the slot-and-filler formalism, that is, frames to represent physical objects, and concepts and scripts to guide and control the process of carrying out the reliability and fracture mechanics analysis. This will provide the mechanism for creating a highly structured knowledge base, which holds all additional information such as documentation, etc. In addition, the most appropriate reasoning mechanism may be that of model-based reasoning, combining quantitative and qualitative data. This final consideration will make the RISC system a practical system given that one of its primary requirements is to be able to combine reliability analysis results with commonsense data.

The RISC planning problem is mainly of the scheduling category in that constraints are assumed and thus activity planning is required in the first instance. If it does become difficult to find an appropriate solution, then relaxing of constraints can be carried out by considering the problem to be closer to the resource allocation category. The required output is a linear plan, with few dependencies, if any, requiring identification.

A scheduling algorithm that combines the concepts of constraints satisfaction with tree-searching techniques was proposed for the RISC System in this chapter. By allowing several searching algorithms to be applied, several different schedules can be presented to the user from which the user may choose the most appropriate. The form of the tree is described in Chapter 5 and the searching algorithms are demonstrated in Chapter 6.

5 IMPLEMENTATION OF THE RISC KBS

The aim of the RISC project was to demonstrate the feasibility of providing offshore operators with a practical computer tool, the RISC System, for rational inspection scheduling for fixed offshore platforms. This RISC System will need to incorporate and integrate:

- reliability-based fatigue and fracture mechanics analysis modules
- external databases of material properties, and inspection techniques
- operators' own procedures and resource constraints
- geometric and other design data, inspection and analysis results for the structure

In Chapter 4, techniques from applied AI were presented and it was proposed that some of these would be useful in designing and implementing a usable RISC System. This chapter describes the work carried out in the design and implementation of a demonstrator knowledge base system. The demonstrator forms the basis for a complete RISC System for the rational reliability-based fatigue fracture mechanics scheduling of IRM for fixed offshore platforms.

The first task carried out was to identify and specify the requirements for the RISC System. This specification and a knowledge of the available analytical tools allowed functional subsystems of the KBS to be identified. Issues and problems relating to these modules are discussed and algorithms from applied AI techniques are explained. A complete Scheduling Model was then proposed and is described here. This model encapsulates the assumptions made when applying the reliability fatigue fracture mechanics analysis. Design details for concepts and the knowledge sources required for the RISC System are explained and some examples given. A user interface was also designed which defined the interaction between the user and the final RISC System. Finally, a description of the implementation of the prototype RISC System, or RISC Demonstrator, is given.

5.1 REQUIREMENTS OF AN IRM PLANNING SYSTEM

The requirements of the RISC System were based on the information given by the Working Group of engineers from operator organisations. In addition, consideration was given to current procedures to ensure that a pragmatic and workable system was designed and the current IRM procedures were reviewed and described in Chapter 2.

5.1.1 Conceptual Model

The requirements for the conceptual model include what information should be included in the RISC System and what functions and results it must provide. In addition, the RISC System is one that needs to be expandable and extensible. For instance, new information and changes relating to guidelines will be required to be input. The reasoning behind derivation of input data to the analysis, the interpretation of analysis results, and the generation of schedules needs to be made explicit. This allows users to question results and to interact with the system.

These basic requirements suggest a knowledge base system approach be appropriate. The base decision procedure is rooted in structural reliability analysis, which requires much data. Interpreting the analysis results and combining of possible inspection actions into a rational schedule requires many forms of information. The RISC System includes external analysis modules and databases integrated with knowledge bases with information on the use of the modules.

The RISC System was conceptualised as a KBS with external modules, as shown in Figure 5.1.

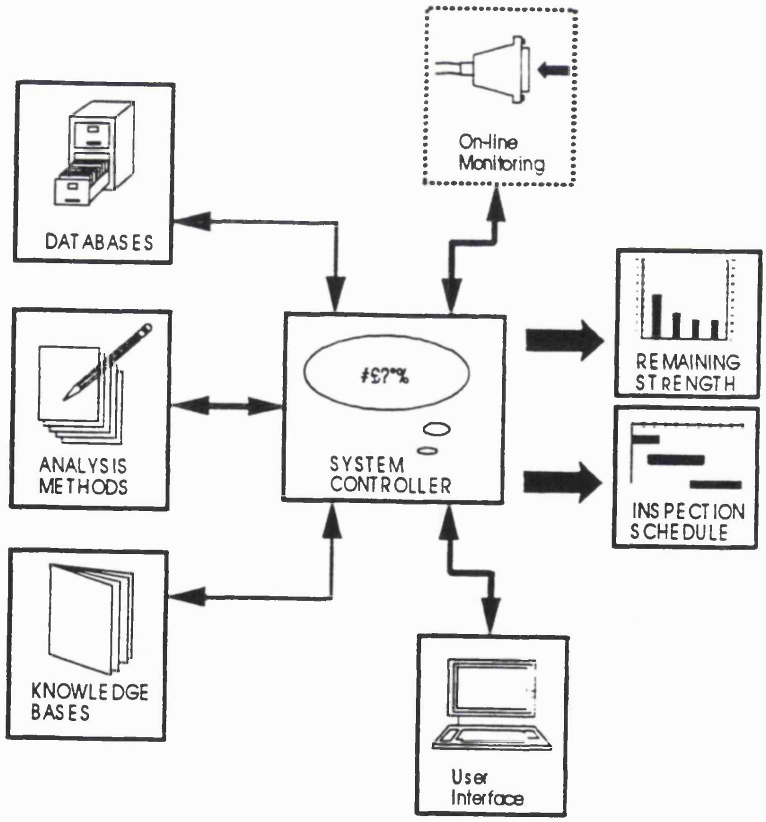


Figure 5.1 The RISC System

The knowledge bases contain information to be used by the analysis and scheduling procedures. In particular, it is the explicit data and information for the structure in question that will enable rational decisions to be made during the production of a schedule. The requirements for these knowledge bases are outlined below.

5.1.1.1 Structural Knowledge Base

The structural model used by the RISC System requires data that can be broadly classified as that arising from design and manufacturing or the results of inspections and environmental data. Design and manufacturing data would include the geometry of the structure at macro and micro levels, design assumptions on environmental loads and conditions, weld details and other information relating to changes made to the design at manufacture and installation.

The RISC System cannot and should not hold all inspection information in detail; only information that will be used by the system in analysis and scheduling should be held. The reasons for this are that computer users tend to believe that all information held by the system is used, which could lead to possible error in interpretation of the results. On the other hand, providing various means of entering data in operator-defined forms is particularly important for data on cracks in welded tubular joints, as there is significant variation in the information recorded by operators as described in Chapter 2. Ideally, the inspection data should be stored in a format closely associated with methods of reporting inspection results, such as direct input from AUTOCAD drawings, written reports, and, in the future, direct input from inspection equipment. Another requirement is to allow data to be stored according to existing or future standards, such as STEP, or STandard for the Exchange of Product data (Bloor & Owen, 1991).

For most non-critical joints little detail is required. Not every weld needs to be listed out and only the outline data on the existence and size of a defect or crack would ever be used (Rivers, 1986). In general, it is supposed that this would be a matter for the data structures internal to the RISC KBS and that detailed inspection data as recorded should be stored to be accessible by the RISC System if needed. Inspection data to be stored for a node includes the inspection technique used, procedural details, such as dates, and by whom, and the defects and cracks found, such as size and location. In addition, it may be of interest to store marine growth thickness as measurements at sample points on tubular members. Other inspections such as after incidents and swim-round surveys would include data on general structural damage:

- type and position of damage found with a classification, which may be unknown in the case of finding debris on the sea-floor damage during a swim-round survey, but the position on the structure of the damage has not been identified
- position of indication of damage relative to the global structure
- procedural details, such as when, how and who found the damage

Environmental and other loading data may be updated with experience gained on the actual weather conditions suffered by the platform. Depending on the existence of on-line monitoring, types of data may include wind and wave loading for the global structure and structural loading on individual joints and members. Data on the loading on the joints may be updated due to changes in the superstructure or changes to the substructure. Updating may be carried out either by direct structural analysis after making changes to the geometrical data or from on-line monitoring data. Records of reported incidents, such as heavy items dropped overboard, can be kept to provide some indication of areas of the structure which require inspection after these incidents (Dunn, 1983).

5.1.1.2 Analysis Procedures

A knowledge base containing information on the analysis modules which are available for use is also required. This KB would include:

- Analysis routes: the expected overall analysis route as defined by the operator and expert engineers
- Reliability Fatigue Fracture Mechanics Analysis and the Loading Analysis modules
- Database access and handling
- Cost Evaluation for Scheduling module

Information for the analysis modules is required on what input is needed and how to derive it, how to execute the module, how to read the analysis results, including, for instance, any errors associated with the results. Once reading of the results has taken place, they are to be evaluated to decide on the future repair and inspection strategies, for instance, or on the time to the next inspection. On updating the inspection data held on the structure, re-analysis must take place in order to update the state of the structure, given the new information.

5.1.1.3 Reliability-based Fatigue and Fracture Mechanics Analysis

The RISC System will carry out reliability analysis which has, at its core, fatigue fracture mechanics routines making use of realistic loading analysis for the prediction of a joint's remaining life. The analysis carried out must take into account the operator's preferences as an operator may require only certain methods to be used at certain times. The input for this analysis is

- joint dimensions including the initial crack size and the crack size at failure, which is derived from information in the structural model and has associated uncertainties
- maintenance plans in the form of a proposed inspection technique, together with its associated reliability from an inspection technique database, a repair strategy stored with the joint, and a range of possible inspection times based on global structural data

- stress concentration factors (SCF) and the stress history information from the structural model and databases
- material properties from a database
- analysis options, which are stored with the joint information, or as global structural data or as default operator's preferences

Loading data is either stored in databases for the structure or provided by loading analysis using nominal loads, loading bias factors, wave height PSD and cyclic rate/frequency and the response or transfer functions for a joint. The output from the reliability-based fatigue fracture mechanics analysis is an initial deterministic assessment of the crack growth, the cost evaluation of maintenance plans probability of failure and reliability measures and reliability sensitivity measures for a joint and over a period of time. A detailed design of the required input data, the interface with the analysis modules and the output reliability data and a description of the how it is dealt with by the system, is given in the Section 5.3 Detailed Design.

5.1.1.4 Scheduling and Planning

The scheduling of inspections and repairs requires information on guidelines and regulations, scheduling heuristics, resource constraints and costs. Cost information can be difficult to obtain, but actual costs are not required, however, since a relative measure of cost is sufficient.

5.1.2 Outline of the Functional Subsystems

The required functions for the RISC System can be summarised as the evaluation of reliability and costs associated with IRM plans for the tubular welded joints of the platform, based on fatigue considerations; interpreting the analysis results; carrying out scheduling and planning and, finally, updating the structural data given the inspection results.

The Reliability and Cost Evaluation Analysis subsystem chooses the most appropriate overall analysis route to follow, given the current state of the structural model and operator's preferences. Input datafiles are created for each joint and analysis module, the analysis module is executed, and the output from the analysis module is redirected to the structural model. Closely linked to the Reliability and Cost Evaluation Analysis is the Analysis Results Interpretation subsystem. This is required to choose some appropriate inspection plans to consider for the component in question. The choices will be dependent on operators' preferences, as well as the regulations and guidelines, and the past history of the node in question. The decision is also affected by information on the reliability of the inspection technique used, which in turn may be affected by the geometry of the node.

The Scheduling & Planning subsystem is required to produce a usable schedule of IRM actions for the complete structure, by combining inspection plans for each joint. This subsystem will provide an initial suggested schedule and generate new schedules which take into account the operator's requirements and procedures. It may provide warnings in the case of joints which are primary or critical, but for which inspections cannot be fitted into the required weather window. It also needs to allow the user to make modifications to the schedule and re-evaluate its utility or cost, on request. The required final schedule will be lists of actions for the scheduling period. Each list will correspond to the IRM actions recommended to be carried out during one weather window or inspection period in the scheduling period. The detailed time-tabling of these actions are carried out by the inspection subcontractors.

The Structure Updating or Observations subsystem is required to aid the user in entering inspection and other data. The data entered includes interpreted inspection data from engineering assessment reports, repair and damage data from platform damage status register and changed properties of individual nodes of the structure, including minor changes in geometry. In addition, it may be necessary to allow changes to be made here to the operator's subjective measures of any node's importance to the safety of the structure, preferences for choice of analysis modules and databases, and to inspection constraints, such as length of weather window. Any major changes to the geometry of the structure because of major damage or repairs can only be input after re-analysis of the structure has taken place outside the RISC System, as certain parameters can only be obtained from external analysis and only via the System Management interface. As an example of this, the measure of criticality of any node, dependant on the geometry of the structure, is obtained through redundancy analysis which is not currently part of the RISC System. In the short-term, an operator's subjective measure representing requirements to always inspect or to always include in analysis, can be used to overcome lack of complete information on the structure.

5.1.3 User Interfaces

The RISC System is intended to be used by the maintenance engineer in charge of scheduling inspection for the platforms in question, but under the control of the operator. Five levels of potential RISC System users were identified:

- Schedulers are the main users of the system. They are in charge of producing an inspection and maintenance schedule for the following weather windows to be agreed upon by the classification authorities. The main requirements for this type of user are ease of use, guidance in choosing the most appropriate analysis routes and in producing an inspection and repair schedule or plan with adequate justification to be submitted to regulatory bodies.

- Offshore inspection engineers supervise the inspection procedure and repair actions. They receive initial reports on the detection of anomalies and require the RISC System to provide them with guidance on entering inspection and damage data and with rapid re-analysis of the structure for immediate advice on repair.

- Repair specialists are usually members of the maintenance department. They are the recipients of information on anomalies, which they then confirm or deny. If the anomaly has been confirmed as a weld-toe crack or defect, then repair specialists carry out the re-analysis of the structure to decide on a course of repair action.

- Maintainers update the information in the RISC System which is dependent on the procedures which the operator organisation carries out. Guidance is needed in changing information which may be highly sensitive. Procedures for storing and recalling knowledge bases and for testing the effect of changes to schedules and analysis procedures as required must be included. These users require a System Management interface.

- System developers change the reasoning and analysis routes available within the RISC System. These users must have very detailed knowledge about the RISC System's internal structure and hence in the first instance will have no special requirements.

5.1.3.1 <u>Interactive Use</u>

The requirements for running the system will vary slightly for each level of user. It has been decided that offshore engineers need not be direct users of the RISC System as it is foreseen that they pass onto the maintenance department details of anomalies found. If confirmed, the maintenance department then enters this information. The expected interactive users are the scheduling and repair engineers in the maintenance department, who have a good understanding of the scheduling process.

A particular user interface is required for the Observations subsystem which includes interfaces for externally held data to be accessed. This interface should be graphical to allow the users to point at the parts of the structure for which data needs to be updated. In order to enter inspection data, the interface should mirror the paper reports used. The design for the user interface should provide a consistent interface which follows industry standards. It should also reflect the underlying data structures, that is, the object hierarchy (Björk, 1991).

5.1.3.2 <u>Reports and Schedules</u>

There are two main reports which will be produced by the RISC System: the schedule of inspection and repairs, and a detailed log on the analysis results including remaining life of joints. Other reports may be required on the system itself. An example of this is a report on the stored data and its structure; another example is on the Scheduling heuristics, to provide the users with background information on the knowledge bases making up the system.

5.1.3.3 Modifications of Knowledge Bases

The ways in which users may interact with the knowledge bases are in viewing stored knowledge, incorporating temporary or user-preferred constraints or heuristics and in modifying the knowledge bases permanently. Maintainers of the KBS may need to make permanent modifications to accommodate:

- changes to the structural KB, reflecting major changes in the structure and results of analyses carried out outside the RISC System
- new analysis methods, including cost functions, to be incorporated
- databases with the operator's own data appended
- changes to regulations and guidelines which may affect constraints on scheduling

All changes to the knowledge bases need to be monitored and checked for discrepancies and logical errors. The user will be informed of any of these errors.

5.2 A SCHEDULING MODEL

This section describes the rational scheduling model implemented by the knowledge base system (KBS) for RISC. The model is based on the decision procedure described in Chapter 3 and in Faber et al (1994). The RISC KBS controls the use of the analysis modules and combines interpreted results to provide a rational schedule for inspection. The details of this process are given here.

5.2.1 Outline of Scheduling Procedure

One of the tasks prior to the analysis is to create a list of joints to be considered as it would be time consuming to analyse all the joints (see Figure 5.2). The KBS will create input datafiles for the reliability analysis by accessing the database containing individual joint data and gathering all other required data.

Cost evaluation analysis is carried out for given maintenance plans for a joint to provide expected costs for each maintenance plan. The KBS interprets the results to give proposed actions for a joint. After interpretation, a schedule is produced which incorporates the proposed actions as well as is possible, that is, satisfies as many constraints as possible.

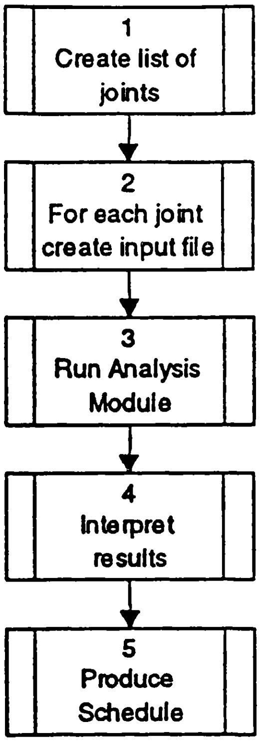


Figure 5.2 The scheduling procedure

5.2.2 A RISC System Session

During a typical user session, the RISC System would be used to carry out cost evaluation analysis on the structure and to support scheduling. The general procedure is shown in as a flowchart in Figure 5.3 overleaf.

As the analysis is carried out on a joint-by-joint basis and analysing every joint in the structure is not practical, the joints making up the structure are ranked. It is important to realise that the ranking procedure is not meant to identify which joints require inspection, instead it provides a priority listing for analysis only. In principle, the user may modify the ranked list to ensure certain joints are definitely analysed. Each joint is then analysed in the order given in the ranked list.

The output from the analysis includes the expected costs for inspection, repair and failure for each joint for each possible inspection period or weather window over the considered number of years which is usually five years. From these results the optimum time for next inspection in terms of expected costs can be deduced.

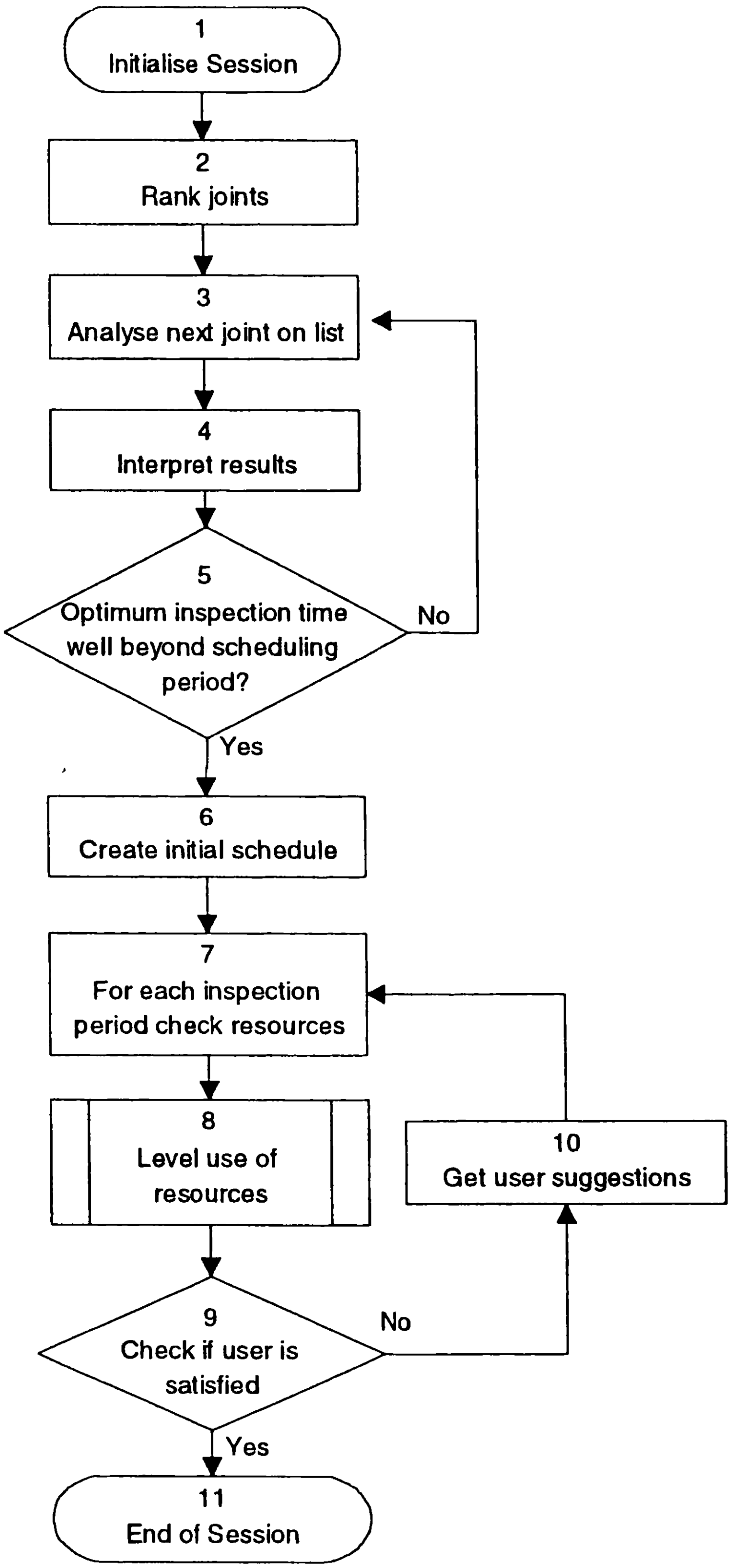


Figure 5.3 A flowchart for a session

A proposed halting condition for the analysis is to stop analysis of joints further down the ranked list when a joint is found to have an optimal inspection time which is long after the end of the scheduling period of interest. This strategy represents the heuristic that all joints of interest are likely to be above that joint in the ranked list. This can be generalised to consider any halting condition, such as "Stop when 100 joints have been analysed".

An initial schedule is created from the optimal inspection plans for all analysed joints. This schedule is modified to take into account the available resources to ensure that the use of resources is levelled across the schedule period. It is at this point in the session that the user's interaction with the system is particularly important. The user will be presented with proposed schedules. If the user is not satisfied or wishes to carry out "what-if" analysis, suggestions as changes to the resource constraints can be made. New schedules are then created based on the suggestions.

5.2.3 The Ranking Algorithm

The purpose of ranking joints making up the structure is to suggest the order of priority for analysing the joints. Analysis then takes place for as many joints as is possible, given the available time for computation, or until evaluation of the analysis results indicates that the remaining joints are likely to have optimum inspection points well beyond the scheduling period of interest.

Several parameters or factors could be used for ranking. In the RISC System, these ranking factors are predefined by the operator. Factors to be considered are

- legal requirements - this criterion will override any other ranking factor
- safety requirements, based on the reliability index (β) given by reliability analysis carried out at set-up and updated according to the latest inspection information
- criticality of joint, based on analyses carried out outside the RISC System
- ratio of exhausted life to predicted fatigue life
- location of joint, including whether joint is a primary member or secondary member
- earlier inspection results

Each factor is weighted and a composite ranking measure is obtained as shown in Figure 5.4. The ranking procedure is applied to all joints of the structure as applying it is very rapid and easy. The normalisation function may be individual to each factor and ensures that the resulting value for the normalised measure lies within the range 0 to 1.

General forms for the normalisation functions $F_D(.)$ for desirable factors, and $F_U(.)$ for undesirable factors, could be

$$F_D(m)=\frac{m-\min(m)}{\max(m)-\min(m)} \quad \text{(a)} \qquad F_U(m)=\frac{\max(m)-m}{\max(m)-\min(m)} \quad \text{(b)} \qquad (5.1)$$

where min(m) and max(m) are either constants or other measures stored about the structure. This $F_D(.)$ ensures that when m has a desirable high value, then n is close to one, similarly $F_U(.)$ ensures that when m has a low value, then n is close to 1. Any factor that the operator may wish to be taken into account is normalised according to this scheme where the maximum and minimum values, whether constants or values provided by databases, are defined by the operator at system set-up. The weights given to each factor are also input at system set-up.

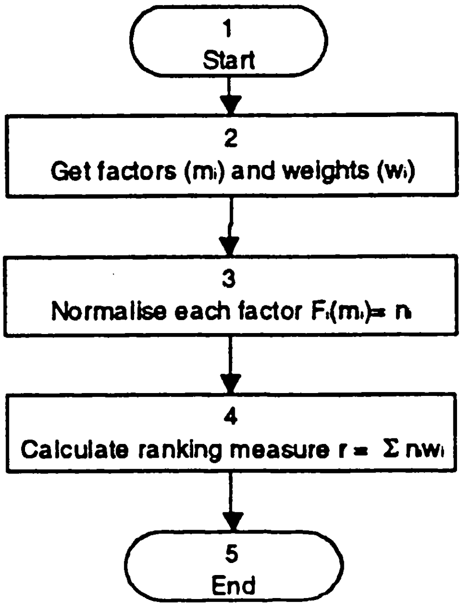


Figure 5.4 Calculation of ranking measure

It is necessary to consider how to deal with factors, such as legal requirements, which if applicable to a joint indicate that the joint must be analysed and/or inspected within the scheduling period. It is proposed that any joint that has a ranking measure at or above a preset threshold value will be analysed and an inspection action proposed for it within the scheduling period. Then the factor's weight is set higher than the threshold value to ensure that a joint for which such a factor is applicable has a ranking measure above the threshold value. This procedure is particularly important for considerations of safety. Alternatively, these factors, which would normally be of a logical type, that is, have values TRUE, equivalent to 1, or FALSE, equivalent to 0, could be considered separately. This is not a favoured solution as it increases the complexity of the ranking task beyond its usefulness.

5.2.4 Analysis of Joints

Usually, the RISC System produces for each joint an input datafile for the analysis module, runs the analysis and interprets the results given in the output file as shown in Figure 5.5. It is possible that analysis is inappropriate for some joints and heuristics need be applied to propose actions for a joint. For example, certification authorities may require certain damaged nodes to be inspected at regular predefined intervals; so for joints on such nodes actions are proposed by applying a simple "If damaged, then inspect every N years" rule, in which case detailed reliability based analysis may be inappropriate for the joint and instead heuristics would be applied. Here the choice and extent of the analysis is explained.

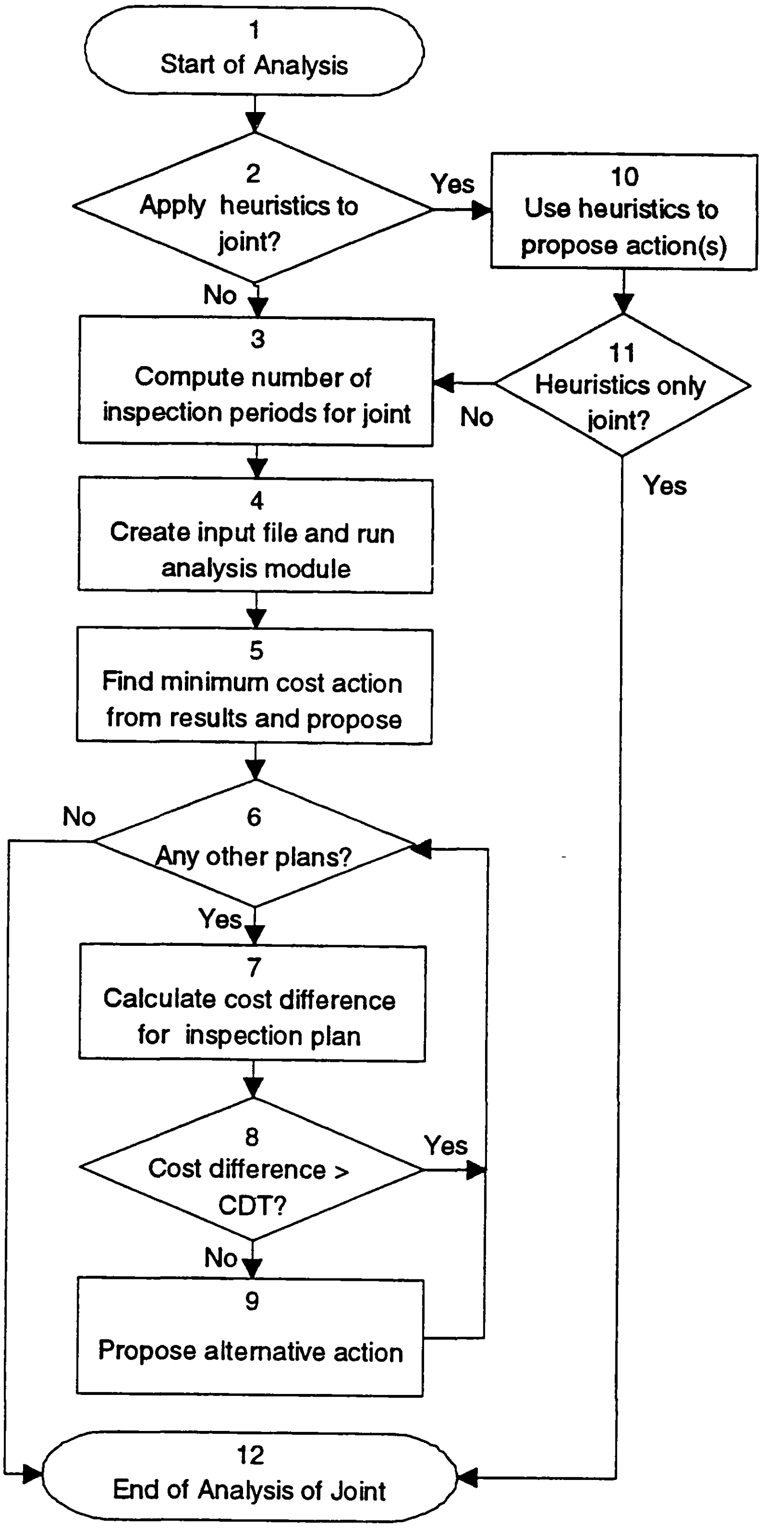


Figure 5.5 Analysis of a single joint

5.2.4.1 Cost Evaluation of Maintenance Plans

The reliability-based analysis module calculates expected costs for various given maintenance plans for a joint. The maintenance plans are given as the set of inspection times to be considered, the inspection technique to be used and a repair criteria representing the chosen repair strategy. Inspection techniques and repair criteria are assumed or derived directly from the information stored on the joint on preferred, or allowable options. The set of inspection times are derived by taking into account reliability.

The number of inspection periods in the scheduling period of interest are denoted here by t_1 to t_N. To be able to gauge whether the optimum inspection time for a joint is beyond the end of the scheduling period, the possible inspection times must be increased beyond the scheduling period, up to t_{N+k}, where k could be set at 5. The default values for the number of inspection periods for consideration in anyone schedule and the value of k as defined above is given by the operator at set-up. As safety is one aspect that should not be compromised, the following rule is applied:

If	the reliability index of the joint falls below its allowed threshold value by inspection period t_n where $n < N+k$
then	the set of possible inspection periods will be t_1 to t_{n-1}
else	the number of possible inspection periods will be t_1 to t_{N+k}.

Figure 5.6 "Inspection times" rule

As an example, consider a scheduling period set to the next five years, each year having one inspection period, t_1 to t_5, and reliability analysis has been carried out over t_1 to t_{10} for two joints as shown in Figure 5.7.

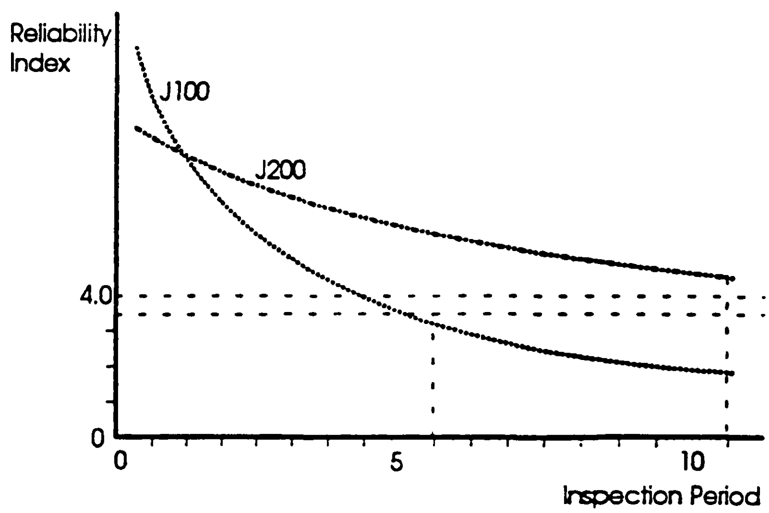


Figure 5.7 Determination of inspection periods

Joint J100 has a reliability index value of 3.6 at the end of t_5, 3.2 at the start of t_6 and a reliability threshold of 3.5. This data indicates that times t_1 to t_5 are all possible inspection periods. Joint J200, on the other hand, has at the start of t_{11} a reliability index value of 4.5 which is above its allowed threshold value of 4.0. Hence for J200, t_1 to t_{10} are all possible inspection times. In this way, inspection actions are proposed only at times that ensure the safety of the structure.

5.2.5 Interpretation of Results

Usually, the output from the analysis module is easily interpreted. The minimum expected cost maintenance plan is selected and this is the proposed action for a joint. If other plans have costs not every much greater than the minimum expected cost plan then these could also be proposed as alternative actions. These alternatives will be considered only if the incorporation of the minimum expected cost plan in the complete schedule is impossible given the resource constraints.

A simple way of ensuring that only a reasonable number of alternative actions are considered is to assume a maximum number for each joint. A second and more rational method of choosing a reasonable number of alternative actions is to set a threshold value for the maximum difference in expected costs between the optimum action and the alternatives. An expected cost difference threshold value (CDT) could be given by

$$CDT = C_Y / K N_{IP} \quad (5.2)$$

where C_Y is the yearly inspection budget, N_{IP} is the expected number of inspections in an inspection period, and K is greater than 1. The actual value of K would be chosen by the operator such that the total number of alternatives does not become too large and once experience has been gained using the RISC System.

5.2.5.1 Heuristics-based Evaluation

In any structure and at anyone time there are likely to be a small subset of joints for which full cost-evaluation of maintenance plans as carried out in the RISC System is inappropriate. This would usually occur if there is no reasonable model or analysis module within the RISC System that could be used to predict the behaviour of the joint. The reasons for this may be that the joints have damage such as buckling, denting, but not fatigue crack growth, or they have had major repairs such as grouting, or clamping. Alternatively, the node may have been defined by the operator or certifying authority as a high priority for inspection, that is the joint is highly critical or primary. Inspections at predefined intervals will be specified. For this group of joints, the operator is likely to wish to analyse these joints in order to determine the effect of forcing inspection at predefined times on the schedule and on the integrity of the structure. The results of this analysis may be used to present a case for allowing a longer interval to next inspection to the certifying authority.

If a joint has been defined as one for which a heuristics-based approach is required, then heuristics associated with the joint on appropriate intervals of time for inspection are triggered to suggest possible actions. The proposed actions for these are specially signalled at the scheduling stage. For "heuristics-only" joints, cost evaluation of maintenance plans is not carried out. Instead, inspection actions for these joints are included in the schedule at the appropriate inspection period. For the rest of the joints, analysis is carried out and all actions resulting from this are classified as alternative actions.

5.2.5.2 Halting of Analysis

Analysis is carried out on the ranked list of joints until the minimum expected cost plan is for an inspection period after the end of the scheduling period, that is, that the time for the cost minimal plan is at t_n, where $n > N+5$, say, and t_N is the last inspection period in the schedule period of being considered. The actual value of n would be given at set-up by the operator. Alternatively, the analysis may stop at some point in the list of ranked joints given by the user or if the number of proposed actions on the joints requires resources which far out weigh the given resources. These two last halting conditions may be set at runtime.

At the end of this procedure, minimum expected cost actions have been proposed for highly ranked joints. In addition some alternative actions are also given for each joint.

5.2.6 Combined Scheduling of Actions

The final stage is to combine the proposed actions into one schedule for the structure as in Figure 5.8. The schedule gives the actions to be carried out at each discrete inspection period and takes into account global resource constraints. An initial schedule is created by combining the proposed actions, that is, by listing the proposed actions for each inspection period within the scheduling period. This schedule has to be modified to take into account resource constraints such as global safety requirements, length of time available for inspection and number of divers, available inspection equipment, location of diving vessel, and guidelines and recommendations.

To carry out modifications to this initial schedule, it is necessary to be able to measure the quality of the new schedule. For instance, a schedule which is unusable is a low quality schedule. Furthermore, in the case where there is a choice of modifications, a measure of quality enables the system to choose the best options.

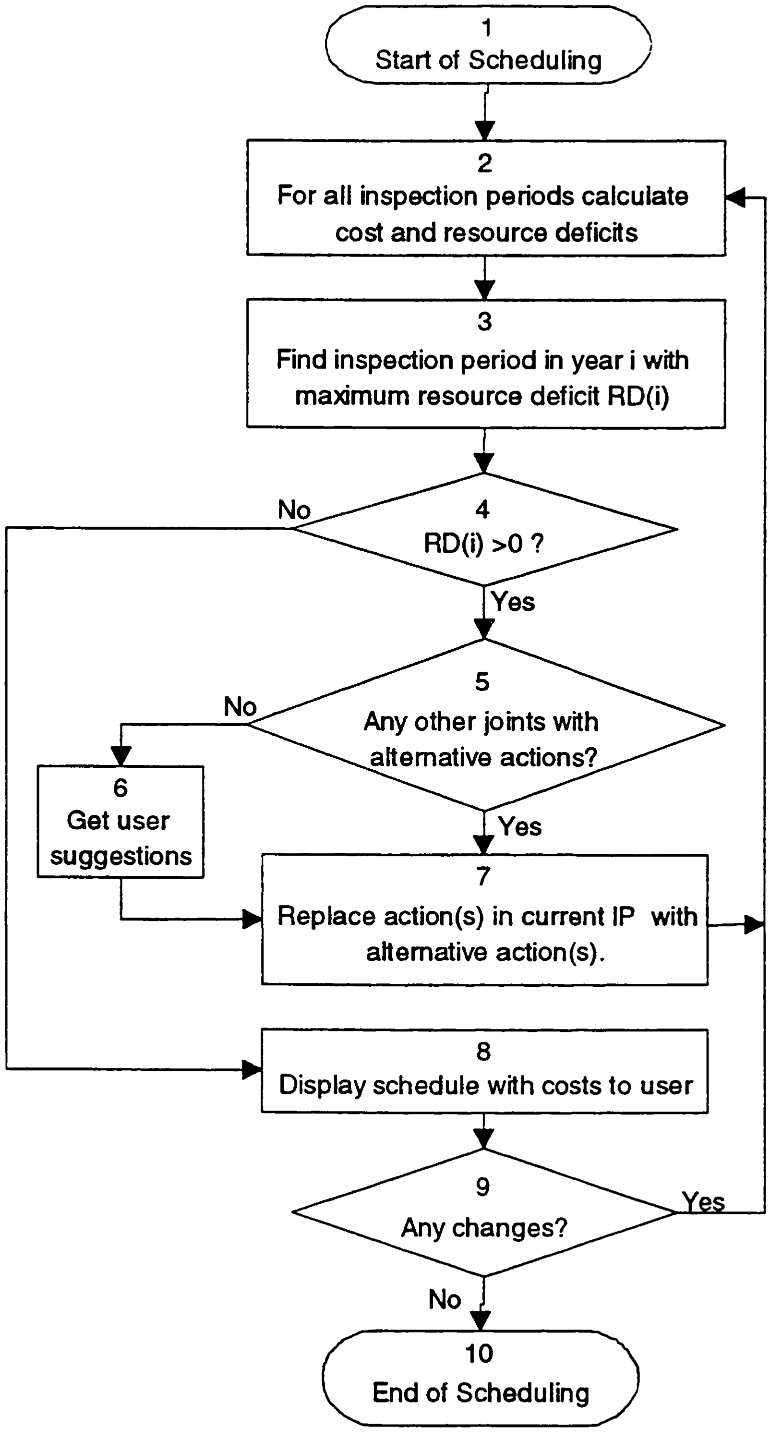


Figure 5.8 Combined scheduling

5.2.6.1 Schedule Utility Function and Resource Deficits

The quality of a schedule may be measured by two related factors: cost and satisfaction of resource constraints. Initially, these two factors may be considered separately. Since the cost of inspection and repairing for each action has already been evaluated, an estimate for the total cost for each schedule is simply estimated as the sum of the individual costs. This cost does not include the added costs of exceeding any scheduling constraints. As there is no direct method of evaluating the exact overall cost for the complete schedule, it is necessary to consider resource constraints.

The satisfaction of resource and regulation constraints is measured by how many constraints are violated. This is indicated by any positive resource deficits. The utility function is then some combination of the resource deficit measure(s), where the resource deficit, $RD(i)$, for inspection period i is given by

$$RD(i) = \text{resources used - available resources} \qquad (5.3)$$

In general this is not one measure but a vector of M measures, each representing a particular resource. For a schedule to satisfy completely all constraints requires that

$$RD_j(i) \leq 0 \qquad (5.4)$$

for all j = 1 to M and all inspection periods i in the schedule. To take into account more than one resource, the user is required to enter which resources are to be used as the criteria for scheduling. A schedule utility function S( ) can also be defined in the same way as the ranking function, with each resource having a particular weighting and each resource deficit being normalised, but here allowing values in the range [-1,1]. Thus the value of S( ) is used as a rapid method of deciding between schedules, but closer consideration of schedules, their associated costs and sensitivities must be carried out by the user to make a final decision.

The $RD(.)$ vector function is used for levelling the use of resources to ensure a rational use of resources. Initially the RISC System will only consider levelling of resources if both

- any inspection period has too many actions assigned to it (i.e. RD(i)>0)
- the total resources required for all actions to be carried out is less than or equal to the total amount of available resources over the scheduling period.

At the end of this procedure, a schedule with no (if possible, if not, then fewer) inspection periods with RD(i) > 0 is output.

This procedure has been implemented as a searching mechanism that generates schedules which satisfy resource constraints.

5.2.6.2 Constraints Satisfaction Scheduling

Constraints Satisfaction (CS) scheduling is the combination of inspection plans for joints ensuring that constraints are satisfied. This is carried out by considering alternative actions in the initial schedule, until all or most of the constraints are satisfied. To perform the scheduling task in a consistent and controlled way it is necessary to implement algorithms which combine the search methods and constraints satisfaction concepts from applied AI described in Chapter 4 (Dean & Wellman, 1991; Stefik, 1981; Tang et al, 1994). The method involves structuring the scheduling objects into a search AND/OR tree and then applying simple brute force searching techniques to obtain solutions. The tree is constructed as is shown in Figure 5.9.

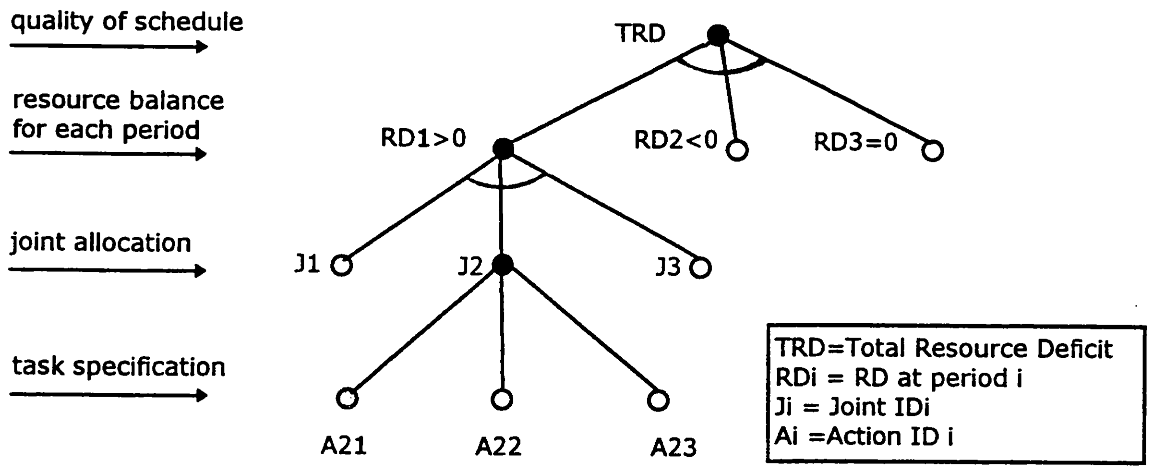


Figure 5.9 RISC constraints satisfaction scheduling tree

The root node indicates the total resource deficit (TRD). The scheduling procedure decomposes the problem to allow parts of the schedule to be considered in turn and hence the next two levels are AND nodes. The first set of nodes below the root represent different action periods and each has an associated resource deficit value. Below each of these are the nodes representing components, here the joints, to be inspected. Each component in turn has a default action as well as a number of alternative actions and these are the leaf OR nodes.

The search tree structure is constructed while assembling the initial schedule. The task of adjusting the initial scheduling then becomes a problem of systematically searching for alternative actions that are consistent with the current resource constraints to reduce the resource deficit. The techniques employed are breadth-first, depth-first and best-first methods. The best-first option was implemented as a heuristics-based search to allow a generalisation of the criteria for searches. This is explained in greater detail in Chapter 6 when example schedules are given. At the end of each search, the tree should be balanced so that $RD(i) \leq 0$ for each inspection period ti and an action is chosen for each component.

5.2.6.3 Neighbouring and Similar Joints

After levelling the use of resources, it is convenient to consider if any other joints should also be considered for inspection. This is desirable only for those inspection periods that have not had all of the resources used up by the already scheduled actions. These additional joints may be neighbouring, and hence it would be convenient to include these. Additional joints may be similar joints to those already included for inspection so should also be of concern. For these, actions may be added at the appropriate inspection period. The inspection and expected repair method will be derived from the available equipment in the inspection period and any stored joint information. This procedure is controlled by the user.

5.2.6.4 User Modifications

At the end of the searching process, the user has a selection of feasible schedules from which to choose a final schedule. Based on the resulting proposed schedules, the user may wish to make changes to the resource constraints and then run the searches again to generate further proposed schedules. Further, the user may wish to make additional changes to the generated schedules by making modifications to scheduled actions, or by suggesting new actions or selecting more joints for additional analysis.

At the end of this process, a rational combined schedule for the complete scheduling period is output for which total expected costs have been calculated. This scheduling procedure is mainly under the control of the user.

5.3 DESIGN AND DATA STRUCTURES FOR THE RISC SYSTEM

The detailed design of the RISC System took place over several, iterative stages. It was highly dependent on the reliability-based cost evaluation analysis module: only when this was close to completion could the details of the data structures and knowledge sources required for controlling the analysis module be finalised. The following describes the RISC System architecture and the knowledge sources required, before giving designs for the knowledge sources and data structures.

5.3.1 RISC Software Architecture

Having identified the functions required for the Scheduling Model described, it was possible to identify the basic architecture of the RISC System. This software architecture, as illustrated by Figure 5.1 on page 149, consists mainly of

- knowledge bases holding knowledge on the structure, scheduling and the analysis modules
- a system controller that controls the way the various databases, knowledge bases and analysis modules are used to provide an IRM schedule

- analysis modules, or knowledge sources, containing reliability-based fatigue fracture mechanics, the loading analysis and other analytical routines required in the setting-up process, and heuristics representing operator preferences
- databases containing probabilistic data on inspection reliability and on material properties, and deterministic data on stress concentration factors and on costs
- a user interface which is graphical and interactive
- a module to allow new information, such as environmental data from on-line monitoring, to be added to the system

The knowledge bases provide global data structures by which the different knowledge sources or analysis modules interact. When an analysis and scheduling session is started in the RISC System, data of different formats is gathered and converted using the knowledge base to form an internal structural model. The knowledge bases are also used to contain to hold skeleton text files used to generate the input files for the analysis modules.

Knowledge sources (KS) of the RISC System include a number of existing analysis modules and a number of heuristic reasoning support programs. The latter perform the tasks of intelligently selecting relevant joints, evaluating analysis results and combining optimal and rational scheduling etc. KSs do not call each other but are invoked by the System Controller, and the KS read data from the shared dynamic knowledge base. External analysis modules are coupled with rule bases which transfer some of the data in the shared object base into text files in a format that can be understood by the analysis modules.

Users have control of the whole scheduling process and are allowed to override the decisions made by the knowledge sources. A user's decision will be particularly important when the information about the history or the installation data of a joint is unavailable.

5.3.1.1 RISC System Tasks

The RISC System is tightly integrated with the user interface, allowing the users to control the system. The tasks it can carry out are presented to the user as a structured hierarchy of Windows and these are shown in Figure 5.10 overleaf.

The structure of the knowledge sources and the design of the user interface is based on this hierarchy. Users control the performance of the RISC System through an event-driven mechanism.

5.3.1.2 RISC System Components

The main group of knowledge sources are those for the View & Run subsystem containing the analysis modules, in particular RISCREL, and the sets of heuristics and procedures for ranking and scheduling. The identification and the development of these knowledge sources are essential in providing the problem solving power for the RISC System. View & Run subsystem contains three groups of KSs to carry out Ranking, the IRM Analysis with RISCREL and for Scheduling. Other knowledge sources are required to set up, update and maintain the data for the RISC System.

Table 5.1 lists and classifies the components making up the complete RISC System. In the table the RISC System components are divided into five major categories: KB (Knowledge Base), DB (Database), IR (User Interface and other Interface Routines) and AM (analysis modules).

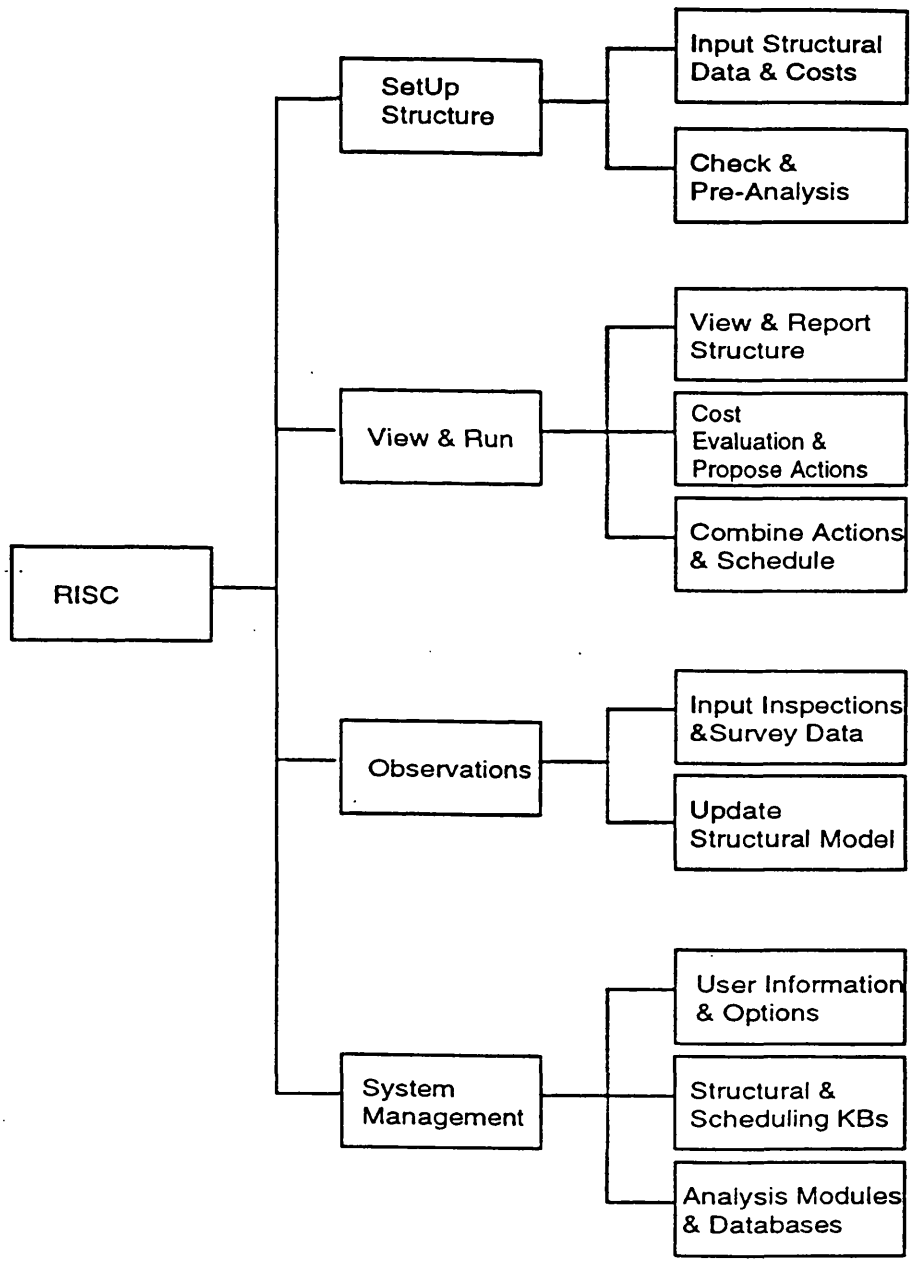


Figure 5.10 RISC hierarchy of tasks

Table 5.1 RISC System components

No	Component	Function	Category
1	System Controller	Overall control of other components	KB
2	Set-Up KS	Sets up new structural model and the KB that stores this	KB
3	Pre-analysis KS	Runs SCF, ULDAN and initial reliability analysis KSs after setting-up of the structural model	AM
4	SCF KS	Calculates or obtains an SCF value for a joint	KB
5	Ranking KS	Selects joints for consideration and cost evaluation analysis	KB
6	Maintenance Plan Formulation KS	Proposes maintenance plans to be considered for a joint	KB
7	Cost Evaluation KS	Calculates expected costs for maintenance plans and reliability indices for a joint by running RISCREL	AM
8	Input Files KS	Creates text input files for an AM	KB
9	Propose Actions KSs	Finds minimum cost action and alternative actions from cost evaluation results file	KB
10	CS Scheduling KS	Generates schedules based on Constraints Satisfaction and different searching mechanisms	KB
12	Observations KS	Runs text file interpretation for entry of inspection results, stores data in internal format and in database and carried out and structural updating	KB
11	System Management KS	Editing functions and KBS checks for maintenance of the system	KB
12	Database access	Runs routines for access to data in databases	IR
13	Structural DB	Structural data is stored as a set of DBs for efficiency	DB
14	Inspection Techniques DB	Contains inspection reliability data for each technique)	DB

No	Component	Function	Category
15	Materials DB	Contains material properties data (Paris' Law C and *m* parameter values)	DB
16	SCF DB	Stores SCF values	DB
17	User interface	Runs the graphical user interface	IR

5.3.2 System Controller

The System Controller is at the core of the architecture of the RISC System and it provides the following basic functions

- control of tasks, which includes managing the use of KBs and storing control data for later reporting to the maintainers of the KBS
- interfaces for the user via a graphical user interface, and for the analysis modules and databases by creating suitable text datafile and reading the text output files
- rule-based reasoning support and rule set control facilities
- error-handling and tracing of reasoning processes

The implementation of the RISC System Controller adopted a task-driven control scheme to carry out the tasks specified by a user through a graphical user interface. In the main, the System Controller controls two different types of knowledge sources, rule-based KSs and analysis KSs.

An internal structural model is used as the data structure for all the knowledge sources, including analysis modules. It is dynamic in the sense that it is in working memory and is modified continuously as the analysis and scheduling progresses. The rule-based knowledge sources use this structural model directly. On the other hand, external analysis modules use this internal model in an indirect way, in that the input text files are created from the internal model. Using text files to provide interfaces to the analysis modules ensures the independence of the architecture of the KBS from the analysis modules. The use of intermediate files is shown in Figure 5.11 overleaf.

5.3.2.1 <u>Control of Rule Set Knowledge Sources</u>

The role of a rule set is to group rules for a particular task and can be enabled or disabled. If a rule set is disabled, it will not be used when the inferencing is started. The reasoning process is more efficient and hence faster if there is only one enabled rule set in the system. The System Controller controls all the rule sets, but at any one time only one is being performed.

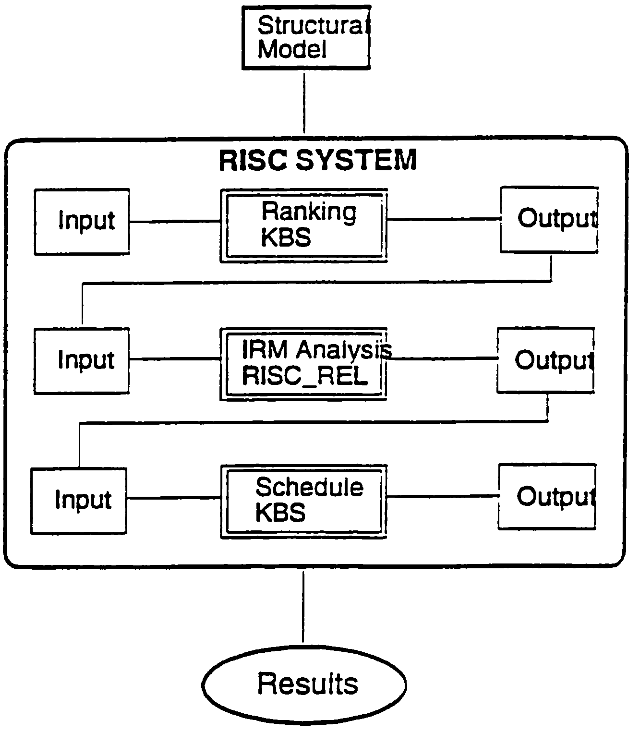


Figure 5.11 The use of intermediate files

If a particular task is to be performed by a rule-based knowledge source, the System Controller enables the rule set associated with that knowledge source and starts the rule-based forward and backward chaining reasoning.

5.3.2.2 Interfaces to Analysis Modules

The control of analysis modules is concerned with preparing text datafiles, running the analysis module and reading of output text datafiles. When an analysis module is to be used, the System Controller checks the preconditions of the task. Preconditions may include that the module should not have already been run, that previous tasks have been carried out, and that an input datafile has been prepared. If the preconditions of an analysis module are confirmed by the System Controller, then the Controller invokes the analysis module, and waits until the results are produced. An analysis module is executed, its output is directed to a text datafile and the file name of the output is fed back to the System Controller.

If the task has already been carried out, then the analysis is not run but the data added to the structural model. This ensures that no redundant work is to be carried out. For example, if the reliability index value for a joint for the required point in time has already been calculated before, then RISCREL is not executed again. If the datafile does not exist or seems incorrect, the process of creating a text file is carried out. The results in the output file are read into the system and stored as dynamic data. This data is then interpreted at a later stage.

5.3.2.3 Database Interfaces

The System Controller can access databases of pre-specified formats. The data is then stored internally to be used as required. For most analysis modules, the System Controller needs to abstract data from databases through the database interface and this data is then stored in the text input files. The access routines for the databases are external module KSs. The databases and the routines have been developed as part of the database tasks.

5.3.2.4 Documentation and Reports

During the whole process of analysis, interpretation, and scheduling, the System Controller maintains the knowledge base and gives users the full access to the information that has already been derived by the KBS system. At any stage of the process, particularly at the end of a scheduling session, a document containing schedules and the relevant information which explain the schedule can be provided by the KBS.

The final reports are a Schedule Summary and Analysis Output. The latter collects together the input and output files for each analysed joint, with a header giving the ranking criteria and weightings. The first contains a list of the joints considered and those scheduled for inspection. The schedule itself is given as a structured table, as in Table 5.2, where the rows are ordered according to the Year of inspection, then by Node-ID. The Joint-ID indicates the weld and model of the node which was analysed.

Table 5.2 Schedule list structure

Year	**List of required IRM resources**		**General region of inspection**	
Node-ID	**Joint-ID**	**Inspection Technique**	**Repair Criteria**	**Comments**

5.3.3 View & Run Subsystem

The main module is the View & Run subsystem in which the analysis and scheduling is carried out. The goal of performing maintenance plan cost evaluation analysis is to identify for the joints considered which maintenance plan is the most rational in terms of cost. The analysis makes intensive use of existing software and it is carried out for a particular joint, producing costs for each maintenance plan and joint analysed. The data flow for View & Run is given in Figure 5.12.

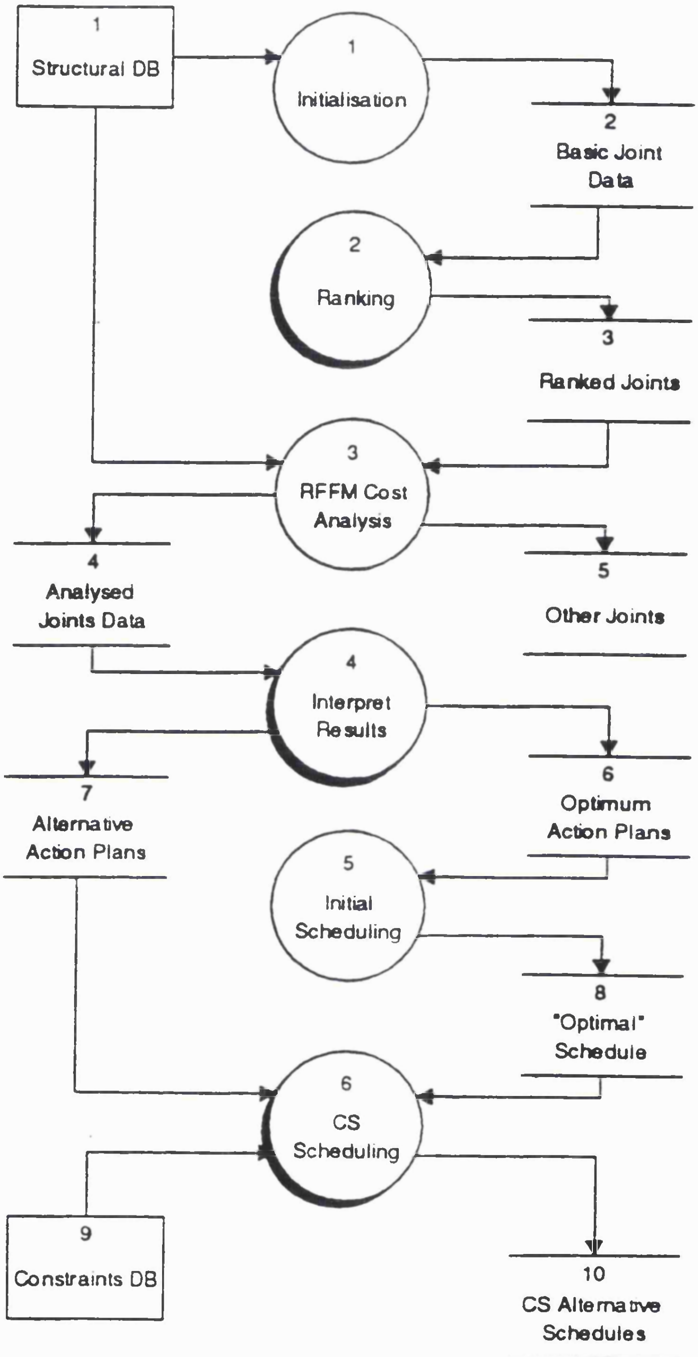


Figure 5.12 Data flow in the RISC View & Run subsystem

The tasks carried out here are

1. obtain general parameters for schedule (e.g. schedule period) from the user

2. select joints to be analysed

3. create input file, carry out analysis, evaluate results for each joint

4. create initial schedule

5. modify schedule to make better use of resources

Additionally, the View & Run module is used to view and query data stored on a structure. The problem solving procedures in View & Run are provided in part by knowledge sources that make use of the existing analysis modules and heuristic knowledge. Rule-based KSs are implemented for each task which may be carried out by the application of heuristics. The advantages of using rule-based approach are that they can be easily understood and therefore can be possibly modified by the domain experts, that is the operators and managers, and they are independent of the KBS architecture and the System Controller. It is also expected that the RISC System would have only a few, that is 10-20, rule-based KSs and hence it is not expected that the interdependencies would become too complex to trace. Because of these factors, rules are particularly appropriate for representing regulations and guidelines. As these requirements change, the appropriate rule-base KSs may be modified without re-implementing the whole KBS system software.

The following KSs with associated rule sets have been developed for View & Run.

5.3.3.1 Ranking KS

As the cost evaluation analysis is computationally expensive, it is not practical for the RISC System to perform an exhaustive analysis for all the joints of the structure. A more intelligent approach, which takes into account the past history and geometric information of the structure is used here to analyse the joints on a selective basis.

The selection or ranking task has been implemented as described in Section 5.2.3. In order to carry this out, data on the joint for each ranking factor is gathered from external databases, text-files on the joint data and from internal data structures.

5.3.3.2 Maintenance Plan Formulation KS

The task of maintenance plan formulation is to choose the most appropriate strategy or set of maintenance plans to adopt for a component in question. The input to this subsystem is the reliability values for selected joint over the scheduling period and information on its past history. This task is carried for all joints in turn prior to cost evaluation. The output from this step is a set of possible NDI techniques, repair criteria and possible inspection periods, which will be used to create an input file for the analysis module. This subsystem relies very much on the heuristic knowledge that operators use when evaluating structural analysis results and making initial decisions. The rules are used in the interpretation process to find possible maintenance plans for any one joint in terms of the allowable times for next inspection, possible inspection techniques and repair criteria for light-grinding versus heavy-grinding or other method. The main work here is in setting the inspection periods for consideration, and rules for this have been described in Section 5.2.4.1.

5.3.3.3 Cost Evaluation KS

The analysis process is also described in Section 5.2. Before carrying this out, maintenance plans have to be generated for the joint and the remaining analysis data gathered from external databases and the joint datafiles. The Cost Evaluation Analysis rule set encapsulate these pre-conditions for running RISCREL and error conditions after running modules. If an error has occurred, the user is notified.

5.3.3.4 Propose Actions KS

This KS in effect carries out analysis results interpretation. The output of the cost evaluation process is a set of cost data for a joint from which an initial minimal cost plan and alternative actions can be selected. The KS identifies possible plans of actions by use of the rules which set the cost difference threshold value as described in Section 5.2.5. This KS also indicates whether or not to continue onto the next joint in the ranked list of joints.

5.3.3.5 Constraints Satisfaction Scheduling KS

The goal of scheduling is to establish how to place actions and resources in order to reduce the costs. The procedure of CS scheduling can be divided into two major activities: to produce the initial schedule based on least cost or proposed actions for each joint, and then satisfy the resource constraints, regulation, and requirements etc. to produce a usable schedule. The first activity is straightforward. The second is more complicated and needs intelligent support to produce the final schedule effectively. Attempting to consider all the constraints and requirements for five years at one time would be computationally too expensive. The strategy employed is to consider the inspection periods with a resource deficit, and the schedule is modified by considering alternative actions which reduce the resource deficit.

The process has been implemented as a scheduling rule set combined with a searching procedure for automatic schedule generation as described in Section 5.2.6.3.

5.3.4 Data Flow, Structures and Storage

The problems of reducing the memory requirements and of representing data are closely interlinked in that use of a representation scheme that allows inheritance of data will reduce the dynamic memory requirements. The data storage problem is explained here and an example of knowledge representation for the RISC System, the structural objects, are now described.

5.3.4.1 Data Storage

One of the major practical problems is for RISC is in memory requirements for both dynamic and long-term storage of data. The knowledge bases are modular which ensures that structured data may be loaded only as and when needed. Furthermore, specialist modules may be re-used as required in different parts of a system. Thus memory requirements are reduced. Traditional flat table databases have been used for storage of data as these can be more efficient in terms of space requirements: when data is required the databases can be easily accessed, the required subset of the databases is then read. The disadvantage of this, on the other hand, is that the data in the databases has to be interpreted as objects to be reasoned within the KBS and this is obviously not as efficient as storing the objects as such in an object-oriented database in the first place. The availability of many industry standard relational database management systems, that is, based on flat tables, and the lack of standard object-oriented database packages obviate this argument against storage of the object data as flat tables. Figure 5.12 on page 174 shows the data flow in the RISC View & Run module, where it can be seen that the structural databases are accessed at two stages: initially for brief data for all the joints in the structure, secondly to retrieve the detailed data required at the analysis stage for each joint.

A KBS can provide flexibility in interaction and in extension of the system. The concept behind KB systems is that the knowledge required to carry out a procedure is kept explicit wherever possible. Thus for instance, should the constraints for scheduling change in time, the architecture is such that it should be possible to make the required changes to the system.

Another important role for the KBS in RISC is in reducing the problem size. The shadowed processes in Figure 5.12 (numbered 2, 4 and 6) reduce the number of components which have to be considered. The Ranking procedure reduces the analysis to be carried out. Interpret Results incorporates a procedure for selecting the most promising (in terms of costs) possible inspection actions for further consideration. The Constraints Satisfaction Scheduling module generates rational schedules from the many possible combinations of maintenance plans, and provides rapid measures of schedule quality.

5.3.4.2 Structural Objects

The basic geometry of the structure consists of a collection of components. Thus the structure object points to component type objects. The basic structural model as a hierarchy of objects is shown in Figure 5.13. Each component of an offshore structure is either a member or a joint which is a welded connection. In the RISC System, for convenience, a further type of component, a node, is identified. The analysis is carried out joints based on detailed consideration of node geometry and loading. Each node is a collection of members and connections, so a node object will point to members and connections.

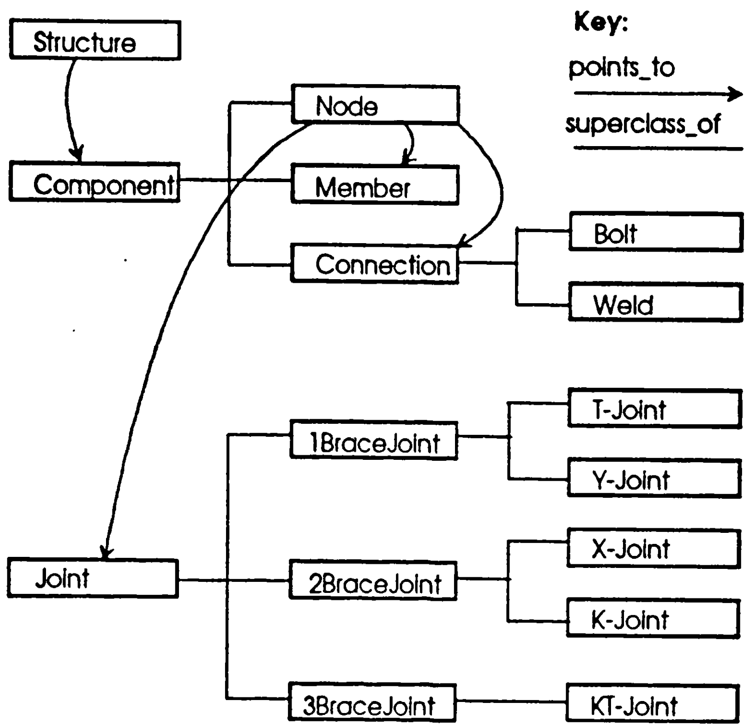


Figure 5.13 Outline structural model

The data held on components can be considered to be granular, that is, the information required varies from a global perspective of relative positions of components to exact dimensions of individual components and welds and materials used. Thus, the structural data is in three layers:

- a beam model of the structure held as point co-ordinates in the node objects
- dimensions and materials of structural members
- connection details such as weld profiles, and materials

Each physical node may be modelled in several forms depending on the loading mode being considered. A node object points to joint objects.

The models of nodes, or joints, can be classified according to the number and placing of the considered braces as T, Y, X, K and KT joints. The allowed parametric equations for SCF calculations depend on the classification of the joint. In addition, the model represents objects representing the concepts required for the scheduling procedure.

Figure 5.14 shows details of the structural concepts and the attributes provide most of the input data for RISCREL within the Cost Evaluation module. The detailed data required to be input into the RISCREL module is given in Chapter 6.

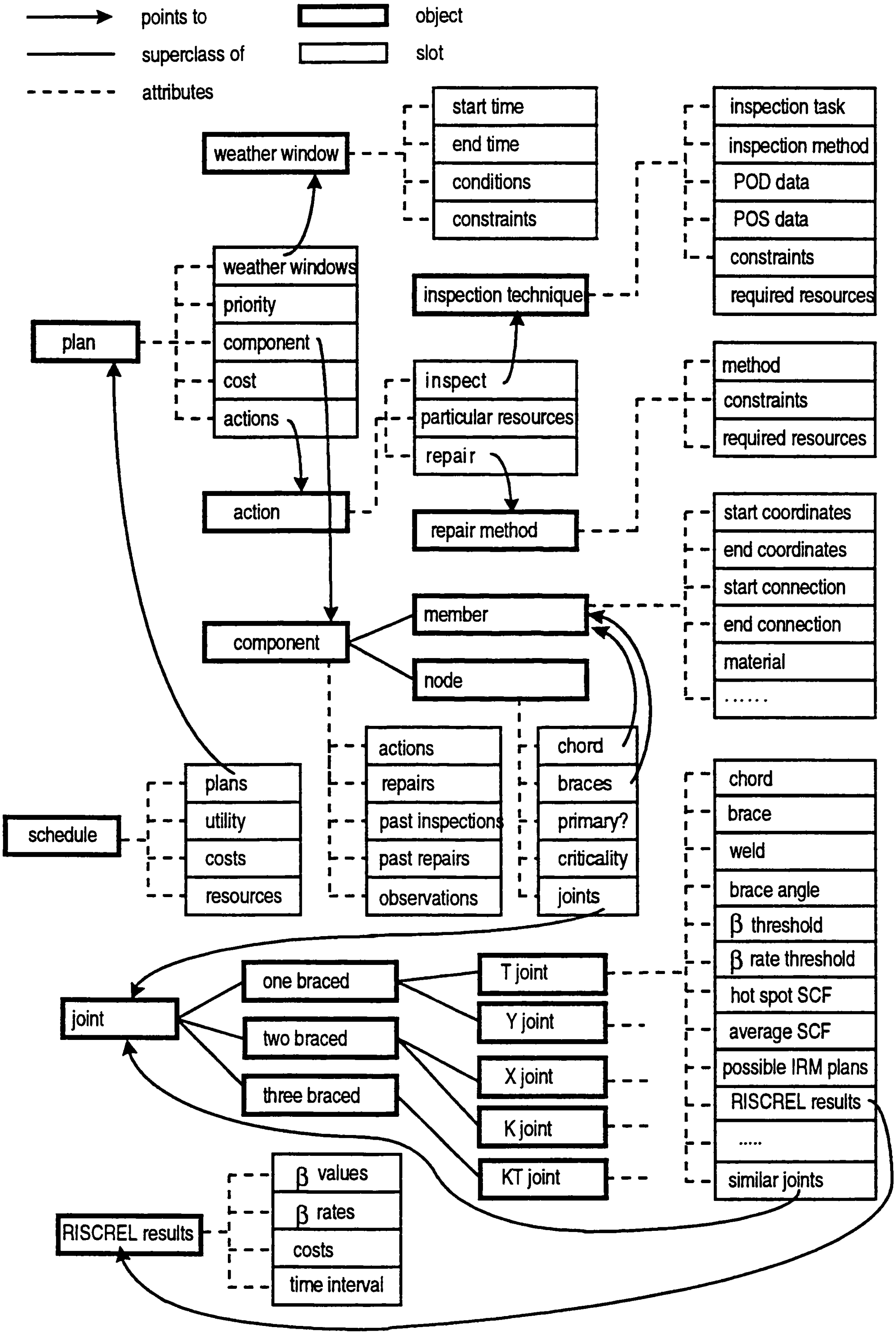


Figure 5.14 Structural objects for scheduling

The figure also shows how the structural objects are linked to the objects related to the planning and scheduling procedure. The raw output from RISCREL on the β values and costs are stored in instances of RISCREL results and a list of these are associated with the appropriate joint object. For each joint analysed there is an instance of the class plan, and this is associated with a set of alternative actions for the joint.

5.3.5 Set-Up, Observations and System Management KSs

Figure 5.15 shows the overview of the process of setting up a structure.

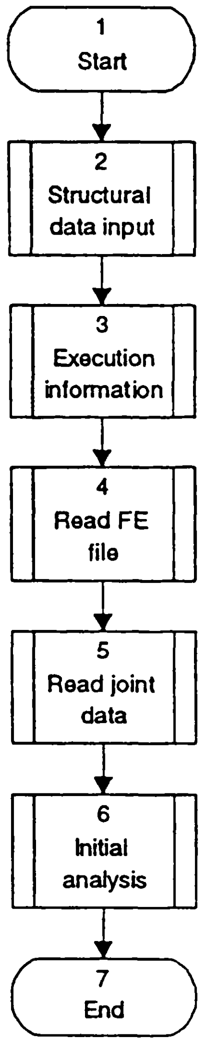


Figure 5.15 Overview of set-up process

Each phase may require collection of information from different sources, such as analysis results obtained by using different computer systems, for example, pushover analysis, detailed finite element analysis, etc. These results will need to be interpreted prior to their use as input to the RISC System. This process is similar for Observations and for System Management. The Observations KS supports the user in entering inspection results and other relevant data into the system, after which re-scheduling may need to be carried out. This subsystem is mainly concerned with providing flexible and interactive ways of entering information into the RISC System so that data of different formats can be transferred from one to another and are held in a consistent and common structure.

5.3.5.1 Set-Up Tasks

Set-Up refers to the tasks that need to be carried out before any analysis of the structure is possible by an expert within the operator organisation. The first task is to set-up the Structural Information. This is platform specific information which is likely to be unstructured, and as a result, the RISC System will extract it from the user interactively. It will include:

- platform data, including name, location and date of construction
- temporal information, that is, the dates of inspection periods, expected dates of weather windows, default scheduling and analysis periods
- resource constraints, such as the number of inspection vessels and of divers permitted to work on the structure, allowed or preferred inspection techniques, initial inspection requirements, etc. which are identified as either *hard*, that is, cannot be modified by the user during a session, or *soft*, which can be modified during "what-if" analysis
- global cost information on inspection techniques, repair methods and for failure, which are employed if detailed costs for an individual joint are not specified

The execution of the RISC System requires certain operator specific information:

- ranking information, that is, factors for ranking of joints, associated ranking weights, desirability of each factor and the maximum and minimum values for normalisation
- cost evaluation analysis information, specifying global analysis options for the structure and default distributions for data

For the analysis, certain data is required for the Analysis Control parameters, Component Fatigue Analysis options, Structural Default values and distributions for joints, cracks, weld and grinding repair. Interpretation of the results requires information on the number of alternatives or the cost difference threshold value to reduce the number of actions to be considered when scheduling. Criteria and heuristics to decide how to search through the alternative actions for a feasible schedule are defined by the user based on a list of priorities. Logistical information on the RISC System, such as, what journal files are to be stored and in what directories and filenames is also required.

Ideally, a finite element (FE) data-file containing the geometric information for a space-frame analysis should be read by the RISC System and from this the connectivity and beam geometry of the structure can be modelled. Alternatively, this data can be read in using a predefined text-file interface. This information is used for displaying the structure as a whole. In addition, the same information is used for identifying locations of structural nodes. It should be noted that Finite Element analysis itself is not part of the RISC System. The FE datafile should contain a node identifier and node point co-ordinates, that is numeric data only. A further file contains the FE node identifier and the corresponding operator-defined node identifier with text information to provide information on connectivity. Each joint will have the following information associated with it:

- Joint Identifier, which specifies one analysis model for a physical node.
- Node Identifier, which is an operator defined identifier and links the joint to the physical node and hence to the location of the joint in the structure.
- Joint Classification, required for the purposes of analysis, by which complex multi-planar joints are split up into simpler, usually single plane joints, based on the load experienced by the joints. This classification is normally carried out during the design phase. It is assumed in RISC that this information is readily available.
- Joint Geometry, which is given as chord and brace diameters, thicknesses, and lengths, and the brace angle.
- Local Weld Geometry, which is sometimes known and, if available, should be stored. This information can be used by some of the stress intensity factor solutions required for the fatigue fracture mechanics analysis. The information stored for each weld is length of weld, and weld profile.
- Material Properties for the material used to construct the joint. The required data are Paris' Law m and C constants. As the RISC System contains a database of material properties, the data is a material DB reference. If actual data for a particular joint is available then a separate feature adds this to the database.
- Loading, which can in RISC be specified as a weighted average stress range (WASR), or as a hot spot stress exceedance diagram in the form of a Weibull curve, or as a combination of sea states and stress transfer functions (SRPD). The sea states information or SRPD is converted to stress exceedance diagrams by relevant analysis modules.

There may be prior inspection results indicating cracks or defects present on the structure. The inspection history is stored in the form crack location, length and/or depth. Repair details may also be stored, with dates and procedural details in text form. Cost information is required for each type of inspection technique, repair method and for failure. Information specifying local analysis options, such as default distributions for data, may be required for the joint. All local data override global options.

Once the basic data has been input for the structure, an initial analysis is performed to set up the loading information by executing, if necessary, the ULDAN module; to find an appropriate SCF value, either from the SCF database or by running SCF routines; and to obtain an initial crack growth curve and base level reliability indices, by initial execution of RISCREL. The final set-up task is to carry out checks on the data to ensure that all required data is complete.

5.3.5.2 Observations Features

In this module the results of confirmed anomalies have to be stored, re-analysis (updating of reliability measures) is carried out and changes are made to status of node, i.e. damaged, repaired, critical, which will affect how the joints are treated in future analyses. The first and second steps are interactive procedures requiring graphic views of the structure. The first requires forms designed so as to reflect the paper reports.

To enter this module, the user is required to enter security data. The main type of information entered in this subsystem relates to inspection results and includes dates of inspection, NDI techniques employed, the interpreted results and repairs carried out. Some joint properties, such as its classification, geometry, weld details, loading data, may also be changed at this point if monitoring provides different data to that at set-up. After new data has been entered, a re-analysis is performed to update loading information, the SCF, and if necessary the base level reliability index and crack growth curves. At the end checks are made on the data to ensure that all required data is complete.

5.3.5.3 System Management Features

In the System Management subsystem, an expert user may make modifications to the user information on the allowed users and levels of users, databases and analysis modules, the report formats, the structural model, and the global operator's information. Only expert users can use this subsystem.

During System Management, the user may modify any data for a structure entered during Set-Up. Information required for the execution of the RISC System may require modification. This information includes which analysis modules and database access routines may be executed and where they exist on the system, what default journal files are to be stored and in what directories and filenames. Any heuristics based on guidelines or standards set by certification authorities are liable to change. Rules embodying heuristics for scheduling are also available to be modified. The default forms of reports and displays should be modifiable to allow new file locations, varying formats and different details of analyses to be stored. At the end of the procedure, checks are made on the data to ensure that all required data is complete.

5.3.6 The User Interface and other Interfaces

Interfaces are required primarily between the analysis modules and the KBS. As rule-based knowledge sources are internal to the development tool, they can directly operate on the internal structural model and so do not require interfaces. In this section, the user interface is also discussed.

5.3.6.1 <u>Datafile Interface</u>

Most of the data for setting up, or entering observations into, the RISC System will be contained in ASCII datafiles. For general platform data and other unstructured information, the user will be required to enter information interactively. Both sources are taken as starting points for the creation of the appropriate datafiles. The use of datafiles is illustrated in Figure 5.16.

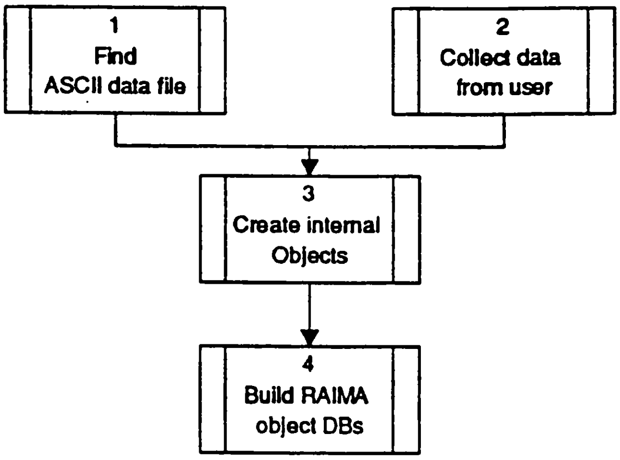


Figure 5.16 Datafiles processing

The data specified in the ASCII text files and by the user will be used by the RISC System to convert to internal objects for the dynamic database. Data for each object is then stored in database tables for rapid initialisation at the runtime stage of the RISC System. Input files required by the Set-Up process are defined as ASCII text files to ensure that data from a wide variety of sources can be used by the RISC System, and producing these Set-Up files are the responsibility of the user. Three text files which need to be interpreted by the Set-Up module are the finite element data, the connectivity data linking the joints to the node, and joint properties and data, representing one model of the node.

5.3.6.2 The User Interface

The RISC System is in effect a decision support system. Hence a feature of the RISC Scheduling Module is that it relies heavily on the user to make final decisions. The rescheduling process, in particular, is highly interactive. Users are allowed to modify a schedule by either delaying or putting forward actions. In any case, the system gives support to users by checking the consistency of the change and calculating the new costs rapidly and displaying the results to the user. This task requires an interactive graphical interface for the user to view the schedule produced and to modify the schedule by making the following decisions:

- re-planning for a joint, which may involve, for example, changing the inspection method or the repair criteria
- rescheduling, by shifting an action from one weather window to another
- planning inspections for new joints which are similar or neighbouring to the joints that are to be inspected
- modifying resource constraints

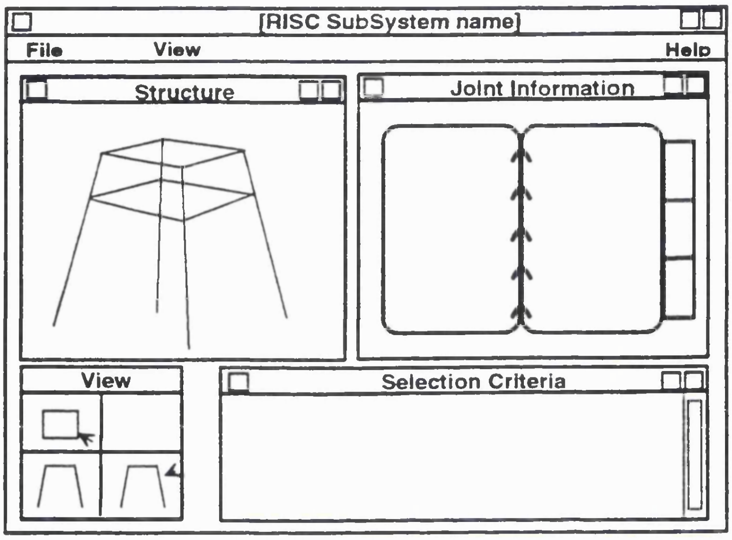
In each of the above cases, an interface is required to aid the user in making a decision. For instance, the system needs to reason about the consequence of the change, report and display the results of the change, give warning messages if an action is to be delayed. In addition, the Cost Evaluation KS needs to invoke analysis modules when an analysis of a new joint is considered by the system as necessary.

A suitable Windows graphical user interface was designed in detail. The structure of the user interface is based on the RISC hierarchy of tasks in Figure 5.10 on page 169 and provides buttons and commands for the user to control the process. Graphical images in the control panel are consistently updated as the system changes status.

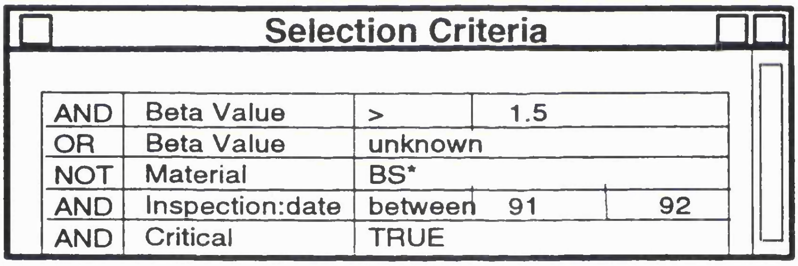
Examples of the designed screens are shown in Figures 5.17 and 5.18 overleaf. Figure 5.17 shows the screen seen by the user when viewing the structural model and general analysis data. The user would need to be able to select components to view according to certain criteria, such as low reliability measures. The selection box allows the user to enter which joints are to be viewed. To start off the analysis and scheduling process, the user is required to input specific analysis control data, such as the scheduling period of interest and the time between weather windows or inspection periods. Some of the control data may be pre-set by the operator during set-up, but the user may wish to over-ride the default values in order to perform speculative analysis and scheduling for future scenarios..

Figure 5.18 shows the screens showing the output from the complete planning and scheduling process, that is, ranking of joints for analysis, viewing the latest analysis output and then the final, interactive scheduling process.

a)



b)



c)

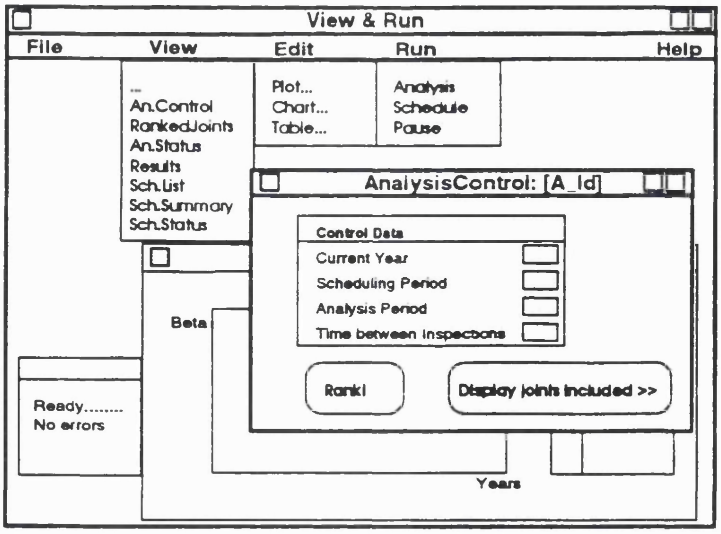
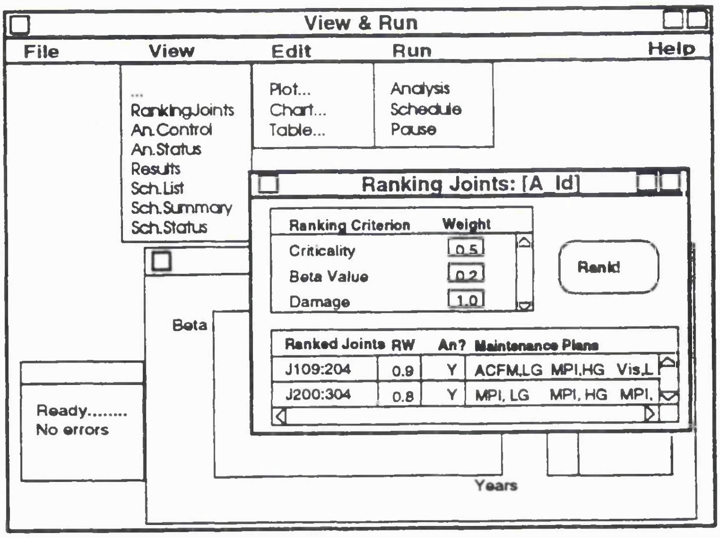
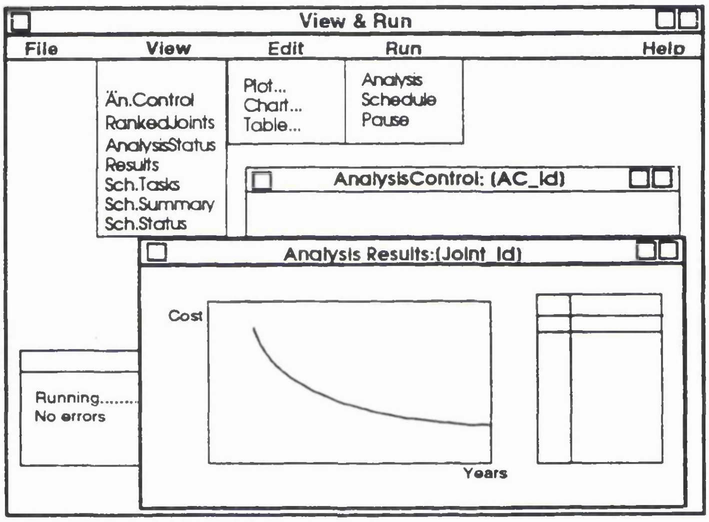


Figure 5.17 User interface (a) Views of structure, (b) Selection box, (c) Analysis control

a)



b)



c)

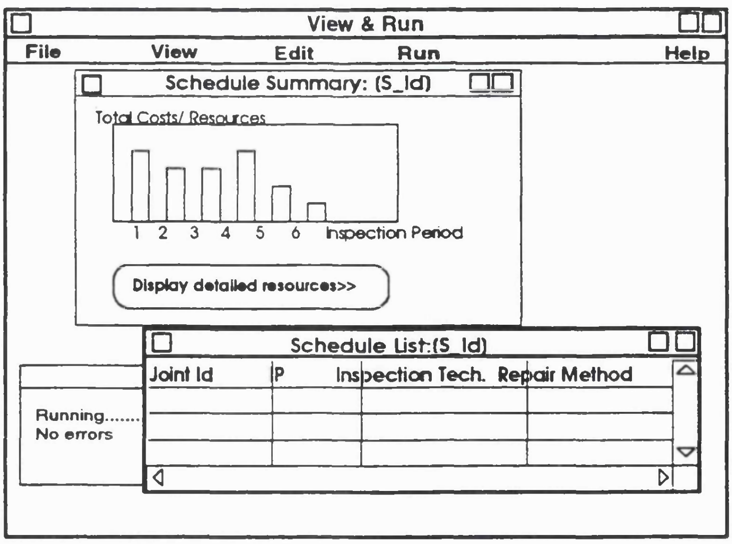


Figure 5.18 User interface (a) Ranking, (b) Output of analysis results, (c) Scheduling

5.4 IMPLEMENTATION OF THE RISC DEMONSTRATOR

For the implementation task, an initial review was carried out of software tools for the RISC KBS development. The packages considered were CASE tools, expert system shells, and database management systems. The choice of tools was restricted by availability for different computer hardware platforms and by requirements laid down by the offshore industry on the platforms for software for the industry.

Following the review, an initial simple prototype was created to aid the design of the object hierarchy described in Section 5.3. Work started on the RISC Demonstrator in parallel to the development of the RISCREL package and as the details of the Scheduling Model in Section 5.2. Some of the details of the implementation of the RISC Demonstrator are given here with some examples of the code from the implemented KBS modules.

5.4.1 Review of Software Tools

There are many different types of software tools currently available: CASE tools, application development systems, languages and libraries expert system shells and the more general AI-based tools. The choice of what combination of tools to use for any software project has become increasingly complex, as it is often difficult to make a clear distinction between their respective capabilities and features. For example, languages are often sold as packaged with standard libraries and specific development tools, such as code generators and debuggers; application development systems may be tied or targeted to particular languages; AI-based tools, expert system shells and application development tools increasingly share common functionality. For the purposes of this section, the following definitions are assumed:

- A *programming language* is a very general tool; it is intended to be applicable to a wide range of problems. Often libraries of routines or other self-contained modules can be obtained which help shorten development time.

- *CASE* (Computer Aided Software Engineering) software development tools often have one or a combination of particular function in the software development cycle, e.g. producing diagrams and documentation for specifications, generating code from specifications, aiding the testing of code, automatic generation of accompanying documentation. These are best used in a commercial environment and are not discussed fully in this report.

- *Applications development systems*, which provide many functions and facilities to the programmer. The interaction between the programmer and the system is carried out throughout the development cycle and hence an application development system is often almost inextricably tied to a language. The reverse is also true in that the use of certain programming languages presupposes an environment.

- An *expert system shell* is a very specific form of an application development system. It is an expert system with the problem-specific knowledge removed, and it is intended to aid rapid development of new expert systems. A shell will have its own very high-level language, though any commercial shell will also be loosely coupled to the language in which the shell was written.

- Easier to define is a *database management system*: this is for the storage and retrieval of large amounts of structured data.

This review is restricted to computing tools particularly suited to implementing AI concepts. It was decided to distinguish between programming languages with or without associated environments, libraries and other tools, other independent libraries and miscellaneous tools such as code generators, expert system shells and problem specific application development systems and finally database management systems. There are many programming language and tool variants, so only commercially available products and fully supported products are considered.

5.4.1.1 AI Programming Languages and Associated Tools

The reasons for choosing a language with any associated tool for engineering software fall into the following categories:

- Purpose of language/tool: For what type of problem is the language or tool intended?

- Speed of development: What features does the tool provide to ease development? And does it increase productivity?

- Correctness and support: Is the language or tool known to be a well-developed product? Is sufficient training and, more importantly, support given? Are there any standards related to the tool?

- Graphical user interfaces: How easy is it to develop a graphical user interface?

Languages which are suitable for implementing AI techniques have been developed both by the pure AI community as well as the computer science community (Baron, 1986). The pure AI languages have their roots in attempting to be able to code programs which reflect the theoretical representation schemes designed by the AI community. The languages developed by this community then can be classified as being either production languages or functional languages. The computer science community, in contrast, are more concerned with developing efficient languages, which would aid them in developing large complex programs. With this aim, object-oriented programming languages were developed. The languages considered are given in Table 5.3 on page 200.

- **Production Languages**

These are the most well-known AI languages which made use of production rules. Requirements for a production language are that the pattern-matching, that is, the mechanism by which a rule is matched against the known or required to decide if it may be applied, should be rich to allow generalised rules and to provide a great measure of control over which rules are used when; and the rule structure be complex enough to provide a flexible knowledge representation scheme. Modern production rule languages allow many conditions and many actions in each rule, such as in OPS5. Some languages, for example, CLIPS and ART-IM, also provide procedural constructs to be included in rules. Finally, reasoning using the rules is usually carried out using forward-reasoning, though backward reasoning is also expected if required. Other examples of production rule languages are OPS83 and Prolog. Production languages do not provide the flexible form of knowledge representation required for engineering software.

- **Functional Languages**

This family of languages were the original AI languages, developed by the first group of people working in the new research field of Artificial Intelligence. To understand the basis for functional languages it is necessary to understand the mathematical concept of a function. A function is a way of relating elements from one set, the domain, to the elements of another set, the range. The range and domain need not be sets of numbers nor need they be of the same type of data. A function may represent a relationship between any form of data or of other functions. In computing terms, a function is a program and hence functional languages allow recursion: the ability to define a program in terms of another program or itself. The classic example is LISP and variants of it, such as CommonLisp, etc. LISP and other functional languages for the development of realistic systems fell out of favour in the late 1980 s, as they do require a great deal of processing power.

- **Hybrid Languages**

Hybrids of the two types of production rule-based languages and functional languages exist. One of the most well-known is Poplog which combines functional and production elements. Hybrid languages never became particularly popular. One possible reason for this is that they tend to have been developed as research tools and hence do not have the features which are associated with languages developed and commercialised by the computer industry.

- **Object-Oriented Programming**

Object-oriented programming (OOP) languages were not developed by the AI community, but the similarity of objects, associated methods and message passing, to the concept of frames, scripts and schemas and their associated daemons, make OOP an appropriate methodology to adopt in implementing an AI frame-based system commercially (Kemp & Saran, 1991). Some frequently-used object-oriented languages are

- C++: This is a very powerful, now much used language in the software industry, which is based on the language C. In 1990 it was the closest to being standardised and thus there were many available libraries of C++ classes to shorten development time.

- Smalltalk: One of the original object-oriented programming languages and a very mature language (standards: 72, 74, 76 and 80). It was also the original windows, icons, mouse and pointer language and thus has probably influenced the present computing world more than any other language.

- Actor: This is intended as an environment for writing Microsoft Windows applications., although it is powerful and has received good reviews.

5.4.1.2 Libraries and Miscellaneous Tools

Libraries of subroutines are commonly available for standard procedural languages. Many commercial compilers and/or environments will provide sets of libraries to be integrated into a programmer's own applications as and when required. Object-oriented programming lends itself very well to the concept of class libraries. These are as the name suggests sets of ready-made classes of objects which can then be used by your own application and thus shorten development time. As for procedural languages, many object-oriented programming languages will come with class libraries, as standard or as an option.

Another class of tool which if used carefully may lead to shortened development times is that of code generators. These allow the developer to "describe", usually through some interactive means, parts of the program. Some details will not be required to be specified and default values are assumed. The generator will then use this "description" to automatically produce the source code which will carry out the task/function required. In the case of implementing AI concepts, the code generators available are part of a packaged AI based development system.

These tools will help implementation of tasks or classes which are most frequently used or difficult to program. Example task and object areas are graphics and user interfaces, such as ready-made windows, dialogue boxes, and databases, with creation and access routines.

- **Class libraries**

As C++ is not a standardised language, class libraries for C++ require careful selection. Some of the considerations to be taken into account are:

- Cosmic versus non-cosmic: A library is cosmic if it has one superclass, that is, all objects in the library share one common superclass and hence there are common properties. If the library is cosmic then the library itself is easier to use and maintain. A disadvantage is that if any changes occur to the superclass, the new library objects may cause a program to behave in an entirely different way.

- Support and continued development of the libraries: This is a common requirement with all other development tools and is more important in the long-term if a cosmic library is chosen.

An example of class libraries is CommonView for developing window and graphical interfaces.

- **Code Generators**

The major problem with these tools is that the code produced is someone else's idea of "good", whether in terms of efficiency or maintainability, code. On the other hand, a code generator can shorten development time considerably, and ensure a minimum standard of code. Most expert system shell and application development system provide code generators.

5.4.1.3 Expert System Shells and Application Development Systems

An expert system shell is merely an expert system without a knowledge base (Mettrey, 1991). A shell will have its own high-level language, based on the knowledge representation scheme, in which to code a knowledge base. A commercial expert system shell is also expected to provide an interface to at least one programming language to allow the incorporation of procedures. Other points to note are that shells often will only provide the control mechanisms and data structures, or knowledge representation, particular to very specific problem types and/or problem areas.

In spite of early predictions that any given shell could and would be used to implement expert systems of all sizes and in all areas, it has been found that in fact most shells are far too restrictive. These restrictions placed on expert systems developers may, only now, be lessening with the changing face of AI-based products. Furthermore, in the case when a software system is developed using some of the techniques developed for expert systems or general AI techniques, it is often over-restrictive to require the system to be able to solve a problem completely and as a human expert would. Non-expert systems may also be very useful without necessarily being able to solve a problem completely. For such systems, it is better to classify them as knowledge base systems.

Reviewing the expert system or knowledge base system tool market is also rather problematic in that the market is in a state of flux. Companies supplying KBS software appear and disappear. Additionally, the rapid changes in hardware have a dramatic effect on the functionality which can be expected from such a tool. Further, the traditional definition of an expert system tool does not cover the more recent AI-based or intelligent application development systems that can be used to develop AI-based programs or systems. Most AI-based systems are not necessarily based on the traditional expert system architecture. They may contain an embedded expert system or systems which do particular tasks, they may utilise searching and data structuring techniques inherited from expert system technology, they may have an element of self-knowledge. The AI-based development tools reviewed are given in Table 5.3 on page 200.

It is also becoming difficult to distinguish between AI-based application development systems and some of the highly sophisticated application development systems. A good commercial AI-based application development system will be expected to provide external language interfaces, user interface development tools, debugging tools, whilst application development systems are now required to provide some unstructured problem-solving typical of expert systems. Some of the considerations to make when choosing an AI-based application development system are

- Knowledge representation and inference: How easy to use is the problem description or data structuring language? And how extensible is it?

- Rules: Are these allowed? How rich is the pattern-matching? And what constructs are allowed (eg, several conditions and actions)? What control is provided over the inference procedure: can backwards as well as forward reasoning be used and can the system move from one to the other? Can the rules be made modular?

- Objects or frames: What ready-defined slots are there? What reasoning is provided? Is there any frame-matching to allow reasoning about stereotypes? What facets are provided? And how extensible and flexible are the frames?

- The development environment: In developing any complex knowledge base system with many and large knowledge bases, it will be important to have sufficiently powerful debugging aids, to be able to view how any one item of information may affect the reasoning carried out. The debugging aids should provide methods for detecting inconsistent information and if possible, allow the developer to see any overlaps (such as items containing similar information) in the knowledge bases.

- Interactive development: In addition, it essential to be able to view the structure of the knowledge bases in diagrammatic form: what "view-points" are allowed? This is not a trivial matter since a good interface has been shown to significantly lower the amount of time spent on development (Jones, 1988). What facilities are provided for interactive development of an application? Or put another way: what functions can the in-built code generator provide?

- Extensibility and add-ons: What tools are available to help shorten the development time? Examples are ready-made modules, libraries, built-in procedures. This would be of particular use for developing the user interface.

- Delivery environments: To deliver software as a packaged system requires the availability of run-time versions of the original system. How easy is it to produce such a package? And what costs are associated?

- Documentation and support: Are there user groups and journals available? Other users tackling similar problems?

- **Production-Rule Development Systems**

Examples are of production rule-based CASE tools are CLIPS and NEXPERT Object. These systems assume that most information is best described as a set of rules. NEXPERT Object is highly extensible and can handle a set of external objects (Aiken & Liu, 1990). It does this by reading an object into memory as a standard item of data and not as an object with links to other objects. This makes it not feasible for reasoning with many objects which are continually up-dated.

- **Object-Oriented Development Systems**

Examples of object-oriented or frame-based CASE tools are Art, KEE, LOOPS, KnowledgeCraft, Kappa PC and ProKappa. These systems provide all three formalisms: rules, objects and procedures, and they generally support multiple reasoning and control paradigms. Yet they do not provide reasoning involving the use of stereotypes to classify objects. This sort of reasoning has not been found by the reviewer in any commercial object-oriented system. Of these examples, Kappa PC is the least sophisticated in terms of the development interface reflecting its PC environment and hence limitations, while the remainder are workstation KBS development systems. In common with most other PC based shells, Kappa PC provides viewing facility in the form of simple knowledge trees showing the hierarchy of objects or dependency between rules.

The first four in the list were originally Lisp-based environments and hence usually require Lisp workstations, although they are now available for some other platforms, whilst Kappa PC and

ProKappa are C-based. ProKappa is in fact an *embeddable* system as it comprises a set of modules which may be included in or excluded from the developers software system as required. This makes ProKappa a powerful base from which to start developing a complex system (Evans, 1991).

5.4.2 Requirements for Implementation

As the RISC System is intended for use in a petrochemical industry, decisions on the implementation of the System were based on the standards described in the Software Integration Platform Specification report by the Petrotechnical Open Software Corporation (POSC, 1991). These specify, in the areas of interest to this work, the use of an operating system complying with POSIX standards, a user interface complying with X-Windows and Motif standards and the Computer Graphics Metafile format for picture description storage.

Other and more general requirements related to development software tools can be summarised as

- Support: RISC was a large project where much of the KBS development was carried out in parallel. Hence the quality of the support which was to be provided was very important.
- Extensibility: It was not always possible at the start of the work to foresee every function expected of the RISC System. Thus a highly extensible set of tools was needed.
- Portability: There are two aspects to this. At the development stage, software development was carried out on a variety of platforms (analysis modules on PCs for example) although all the major work on the System Controller was of course carried out on the chosen development platform. At the delivery stage, it may have been necessary to be able to port the demonstration system onto other platforms.

In conclusion and given the software review in Section 5.4.1, Kappa PC was selected to carry put initial design work for the RISC Demonstrator (IntelliCorp, 1992). For the major part of the development of the demonstrator system, the C-based knowledge base system development tool, ProKappa, (Intellicorp, 1991) was selected. A standard DBMS, RAIMA, was also to be used for external databases (RAIMA, 1991).

In view of the POSC requirements, the chosen development platform of the complete KBS was the IBM RS/6000 with AIX operating system. Parallel development of the analysis modules and database access routines was carried on a variety of other platforms. ProKappa was available for a limited range of computer base hardware at the start of the work, beta version was supplied for the IBM AIX Window environment. Development of the RISC Demonstrator including the

Scheduling Module was carried out using ProKappa on the IBM RS/6000 platform.

5.4.3 The KBS Development Tool, ProKappa

The CASE tool ProKappa fully supports both object-oriented programming (frame-based representation, methods and daemons) and rule-based programming (forward chaining and backward chaining). These two paradigms are unified in a high level environment within which the knowledge base and knowledge sources can be developed, either using a higher level language called ProTalk or using C. In both cases, ProKappa provides good supporting facilities for quick prototyping and software debugging and maintenance. Other languages can be used as external modules to be called as necessary with a suitable interface.

In ProKappa, a class hierarchy with object instances may be defined to describe a complex system. Within the classes and objects, slots are used to characterise the relevant parameters, and methods are defined to express their behaviour. Slot options and monitors are used to impose constraints on parameters. ProKappa also has an active graphical image system and an interface workbench for developing graphical user interfaces. This was used for the prototype user interface.

In addition, ProTalk, a high level rule-based language, is built into the ProKappa system and this was used to develop the RISC System Controller and the rule-based knowledge sources. In ProKappa, rules can be invoked through functions and methods; functions can be triggered and messages can be sent from rules. This makes it easy to embody both procedural and heuristic knowledge into rules to carry out complex tasks. As example of ProTalk rules, consider the Maintenance Plan Formulation KS. In the initial prototype, the rule set object EvaluationRuleSet performed the task of deciding if it is necessary to consider inspections at all based on reliability considerations only. The System Controller triggers theses rules by a C function call, ForwardChain(EvaluationRuleSet). In ProTalk syntax, as expected each rule has two parts, the precondition part (**If**) and action part (**then**). The two examples of rules in the EvaluationRuleSet given in Figure 5.19 and 5.20 are written in ProTalk syntax.

In the NoActionRule, the first line of the preconditions binds variable ?Joint to the SelectedJoint stored in the Control object. In the second line, a message is sent to the object ReliabilityResults which has an attached method function called FindBetaValue! to obtain the value of the reliability index. The third line compares the found β value with a threshold hold value defined in the Joint object. If β in the current year, that is the year under consideration, is larger than the threshold value, the statements in the action part of the rule is carried out. In this case, a NoAction symbol is assigned to the slot Action in the Joint object through the statement ?Joint.Action = NoAction. The function InformUser is called to notify the user.

```
fcrule NoActionRule in EvaluationRuleSet priority 1000
{ if
?Joint == Control.SelectedJoint;
?BetaValueOFJoint = SendMsg(?Joint.ReliabilityResults,
FindBetaValue!, Control.CurrentYear);
?BetaValueOFJoint > ?Joint.BetaThreshold;
then
?Joint.Action = NoAction
InformUser(NoAction, ?Joint);    }
```

Figure 5.19 NoAction ProKappa rule

```
fcrule InspectRule in EvaluationRuleSet  priority 900
{if
?Joint == Control.SelectedJoint;
?Critical = ?Joint.Criticality;
?BetaValueOfJoint = SendMsg(?Joint.ReliabilityResults,
FindBetaValue!, Control.CurrentYear);
?Critical == Yes;
?BetaValueOfJoint <= ?Joint.BetaThreshold;
then
?Joint.Action = InspectNow;
?Joint.InspectMethod = FindInspectMethod(?Joint,
InspectNow);
InformUser(InspectNow, ?Joint);    }
```

Figure 5.20 InspectNow ProKappa rule

The second rule suggests an InspectNow action for a selected joint if the reliability index value of the joint at the current year falls below the threshold value and the joint is a critical one. These two rules demonstrate the integration of rule-based programming, functional programming and object-oriented programming in the ProKappa tool.

Future versions of ProKappa would allow executable independent KBS system to be developed and distributed on different platforms.

5.4.4 The RISC Demonstrator

The RISC Demonstrator confirmed the feasibility of combining analysis modules within the proposed KBS structure for the complete RISC System, as illustrated in Figure 5.21.

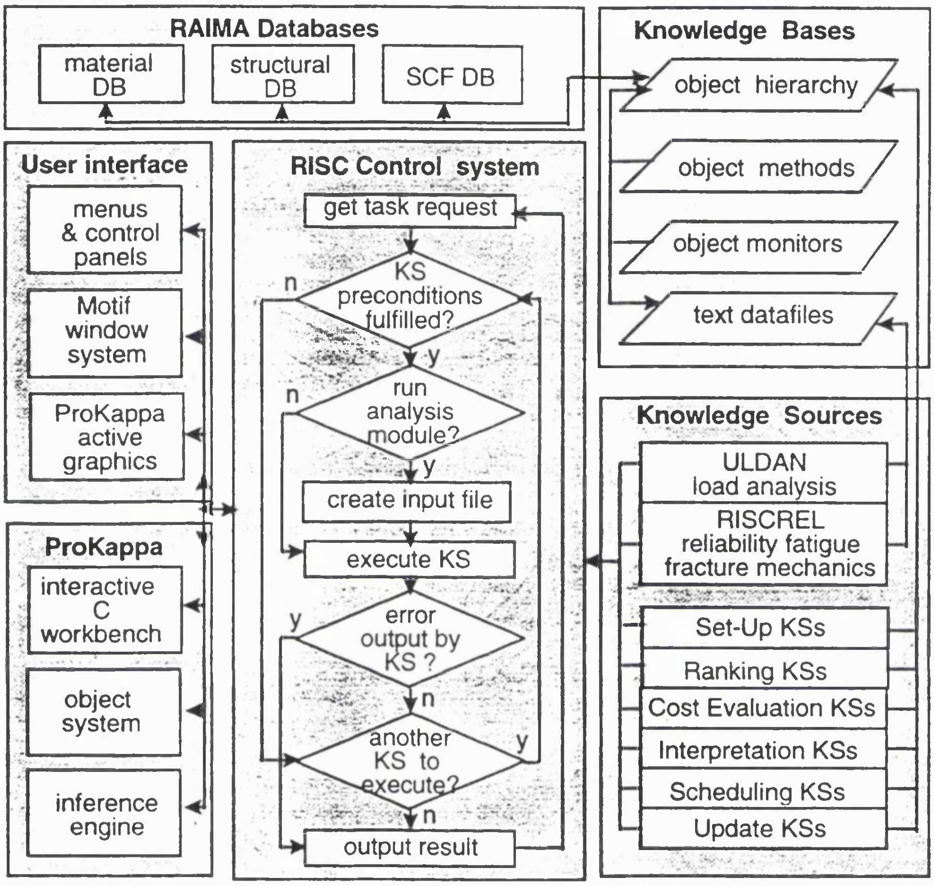


Figure 5.21 The complete RISC System

Implementation of the prototype for the RISC KBS, the RISC Demonstrator, was carried out in parallel with the work on modifications to the structural reliability analysis modules. The reliability analysis modules were combined with the FACTS and CFA limit states to provide a first version of the RISCREL analysis module, COMREL (COMponent RELiability system) which carried out reliability based fatigue fracture mechanics analysis per joint only. COMREL was later extended to produce RISCREL which carried out both reliability based fatigue fracture mechanics and cost evaluation of maintenance plans. For the RISC Demonstrator, COMREL was integrated into the ProKappa system and combined with the Ranking and Execution KBS modules with the use of the ProKappa system. COMREL is run as a separate process and its execution is controlled from ProKappa. The complete RISC System would control execution of RISCREL instead.

The Ranking module and CS Scheduling modules were developed and the basic design for these tested. Simulated RISCREL output files were used to demonstrate displays to the user. assumption of the availability of expected costs per maintenance plan. The CS Scheduling module was developed. The design for the CS Scheduling KS was tested. This work demonstrated how the RISC System containing all the required components could be constructed. Screens from the RISC emonstrator showing a schedule in preparation are shown in Figures 5.22 to 5.25 on pages 207 to 210. The first shows the output from the Ranking module. The next shows the output from three runs of RISCREL for one joint and three different inspection methods displayed by the KBS prototype user interface. In the third screen, an initial schedule is displayed for which too many inspections are planned for year 3, as shown by comparing the required resources (in green) to the available resources (in white). The user may choose one or all possible searching methods to find schedules that are usable, that is that satisfy the constraints. The user may investigate what happens if the resources are distributed differently across the weather windows. Based on the initial schedule, the user would probably decide to change the resource distribution so that no resources are allocated to year 2 and the resources for year 3 are doubled. The output of a search, a usable and rational schedule, is then created, as shown in Figure 5.25.

5.5 SUMMARY

In this chapter, the specifications for the RISC System, including the data structures, that is object hierarchy, and the user interface, were given. The Scheduling Model developed for RISC was described. The scheduling procedure, in effect the main function of the RISC System, can be summarised as follows:

1. Ranking or selection of joints to be analysed
2. Cost Evaluation analysis of joints and maintenance plans
3. Interpretation of the analysis results to provide an initial schedule based on the least cost maintenance plans for each joint
4. Scheduling which makes modifications to the initial schedule to provide a rational schedule satisfying resource constraints into account

A review of available software tools was carried out and from this a set selected for the implementation of the prototype of the RISC System KBS modules. The RISC Demonstrator was developed using ProKappa on an IBM RS/6000 platform and includes modules for the Ranking, Evaluation and CS Scheduling procedures. Its use is described in Chapter 6.

Table 5.3 Software packages and languages reviewed

AI PROGRAMMING LANGUAGES

Name(+Man.)	Plat.(+Price)	Description	Structures	Interfaces	Development	Comments
Actor (The Whitewater Group Inc. Dist. by NEOW Ltd)	Professional version (inc. ObjectGraphics + W.R.Tkit) for PC AT, PS/2 286 +Win 3 (£375-645) (+1MB for W.R.Toolkit or Wintrieve)	OOPL for Windows 3 app. dev.	Objects ObjectWindows (windows classes) ObjectGraphics	C Porting to C++, Pascal	Interactive env. works under Windows Support for DLLs Own lang. Actor Integrated debugger. Compilation: uses token threading	Also available is the Whitewater Resource Toolkit (£145) to interactively modify Windows resources and Wintrieve Reviewed as being more productive than C++ [Evans]
Borland C++ (Borland International (UK) Ltd)	DOS and Windows 3 £300 +VAT	OOPL			Support for MDI, DLL, and DDE Includes Whitewater Resource Debugger	Borland products are known for being nice to use for small applications development. It is also rumoured that Borland C is faster than Microsoft C! Coming soon: Borland's own GUI library.
Eiffel (Applied Logic 081-780 1088)		OOPL	multiple inheritance		Automatic compiler source level debugger auto doc. tools graphics tools Libraries (inc. X-Windows)	
Enfin (? Dist. by QA Training)	MS Windows Dev. kits (£?) OS/2 Dev. kit (£3500)	OOPL / 4GL for Windows 3 app. dev.	?	?	SQL support Editors: database design, user interface, maths models and graphical reports Debuggers etc. Code generator	

Glock C ++ (Glockenspiel marketed in the UK by QA Training 0285 655888)	DOS, OS/2 (£295) PC Unix (£840) RS/6000 station +AIX (£1050-£4200) RS/6000 server (£1540-£6300) Sun MIPS DECstation VAXes, etc	OOPL	Objects		AT&T C++ specifications Strong type checking Support includes all upgrades, guaranteed response time (5 days), example CommonView code and journal.	Recommended by Microsoft until MS have themselves produced their own OO C (C V7!).
LPA-Prolog and Prolog ++ (Logic Programming Associates)	PC+DOS Macintosh New 386 versions (includes ++)			MS 'C' MS MASM DBaseIII	Incremental compiler Toolkits available (graphics, hci, databases, +LPA flex) Usual Prolog environment. 386 versions: upto 4 GB memory 32-bit flops	
Microsoft C V7.0 (Microsoft)						This version is expected to be shipped by end of 1991. MS recommend that people get started with Glockenspiel C++ and CommonView whilst waiting. Microsoft C is one of the industry's standards: THE programming language!!
ML						Mentioned as one of POPLOG languages: never heard of it before!!
Objectkit/C++ (Parc Place, UK distributor AI International)						2 sets of supported libraries + 2 unsupported libraries
POPLOG (Marketed by Integral Solutions Ltd Maintained by Sussex University)		Int. prog. environment		POP-11 Prolog, Common-LISP ML	Provides access to all languages with which it is said to provide interfaces.	

Prolog-2 (Expert Systems Ltd)	PC+Windows 3 (£?) Personal-(£?) Programmer (£?) Professional Plus (can be used on any PC)(£?) 80386 (£?)	Logic prog. language + environment	Horn Clauses (Rules) Data-driven (Forward)	'C' MS Fortran Pascal assembler dBase III + spreadsheet files direct access (Win 3 DDE)	Personal -interpreter only. Rest - interpreter + environment Prof. Plus - compiler = packager 32 bit addressing with VM support Multi-window dev env. Window debugger (all versions)	Can be embedded. No runtime fees. (Also versions for Sun, Apollo, HP/9000, RS/6000)
Prolog++/386 (LPA Ltd)	See above					
Smalltalk (Digitalk Inc. Dist. by Cocking & Drury 071-436 9481)	DOS (£85) 286 version, Macintosh (£140) MS Windows, OS/2 PM (£330)	OOPL! Smalltalk V/Windows is a Windows 3 app.dev.			Integrated debugger. Incremental compiler. DDE	The original OOPL! Smalltalk V/Windows reviewed as more productive than C++ or Actor. See [Evans].
ToolBook (Asymetrix Dist. by NEOW Ltd)	286 PC + Windows 3 + 2.0 MB +mouse+ VGA,EGA etc (£310)	OO toolkit	Scripts to handle events Hypertext	dBase III	OpenScript - own language DLL support Uses DDE	
Zortech C++ (Zortech Dist. by System Science 071-833 1022)	DOS + Windows 3 (£250) Developers - DOS, OS/2 (£400) Science & Engineering - floating point, M++ array language (£600)	OOPL			Library source included in Developers version and C++ tools	

EXPERT SYSTEM SHELLS/APPLICATION DEVELOPMENT SYSTEMS

Name(+Man.)	Plat.(+Price)	K.Repn.	Reason.	Interfaces	Development	Comments
ART						Automated Reasoning Tool ART for Information Management-ART-IM See [Mettrey].
CLIPS		Production rules	Back and forward	C		C Language Integrated Production System See [Mettrey].
G2 (Gensym Corp, US marketed in UK by SIRA 081-467-2636)	UNIX workstations	Features in G2V2.0 include transient objects.				Real-time E.S. shell. G2V2.0 offers multilingual facilities.
Goldworks II (Gold Hill Inc Support provided by AI Ltd)	PC-AT PS/2 MacII Compaq 386 Sun workstations	Frames Objects Rules	For + back	Win 3 -DDE dbaseIII MS 'C' Lotus 1-2-3 Common Gold LISP	Graphics layout tool Menu-driven int. ASCII parser for external text files LISP only with Dev. Interface	
GURU (Micro Data Base Systems)						No longer available. Mark Leaning in Statistical Science has a copy (x3638)

Kappa-PC (IntelliCorp UK 0962-735348)	IBM PCs	Frames (or objects) with certain predefined demons. Rules Functions Single Inheritance only.	Backwards Forwards etc.	'C'	Windows V3 based environment (only- not particularly flexible) Some graphics Own language KAL	Presented as sister to ProKappa but it is NOT compatible with ProKappa. Reviewed as being very slow. Version 1.2 now available with improved examples. **Suitable for prototyping.**
KEE (IntelliCorp UK)	Sun4/SPARC HP RS/6000(Beta)	Frames	Truth Maintenance System (!!) and KEEworlds for comparing alternative scenarios	?	Development tools include: KEEconnection (database interfaces), IntelliScope (db qeuries), SimKit (simulation), PC-Host (for delivery on PCs)	Popular as a research tool. Future of this product is safe according to IntelliCorp, however they are also encouraging migrations from KEE to ProKappa! New booster module KLUE will deliver KEE applications on 286s.
Leonardo (Creative Logic)	Level 2 (£995)& Level 3 versions (£1995): IBM PCs (+OS/2) Level 3: DEC VAX (VMS) Sun workstation Runtime licences - £2000 for 10 Maintenance and support for 2 - £200, 3 - £300, for VAX - 15% of price	Frames, classes Structured rulesets Level 3: certainty factors	For + back Level 3: multi-level inheritance, uncertainty factors and Bayes theorem	Suspend: run any other s.w. Access to stock s.w.(optional): Symphony, Supercalc, Lotus, Dataease, dBase, Btrieve Invoked routines: any language (C, FORTRAN,, Pascal) with parameters	Compiled environment Level 3: dynamic instance and slot creation(!)! VAX/Apollo/Sun: embeddability Graphics Toolkit DBase Interface Lotus Interface Statistics Package Screen Designer with hypertext facility Debugging facilities (logging to file, forced reasoning, etc)	Run-time systems available Packages provide on-site support provided for 2 or 7 days(!!)

Level 5	Macintosh					See [Mettrey].
LPA -flex (Logic Programming Associates)	Mac DOS PC + 4MB EMS 3.2 +LPA Prolog 3 UNIX + DEC VMS workstations +4MB RAM +40MB hard disk+ Quintus Prolog	Frames + demons, constraints, watchdogs Rules Objects	Forward Backward Forward (demons) Uses an agenda to keep track of rules	LPA-Prolog!!	Own english -like language KSL Environment	
M1 (Framentec, France)		Rules Uncertainty factors				Out-of-date info (~1987)
Nexpert Object (Neuron Data 071-408-2333) (Also contact John Kelly at Software Sciences 0252-544321)	PCs (DOS, Win3, protected mode, UNIX, OS/2) workstations(Sun, HP/Apollo, DEC, IBM, NeXT, Sony) minis (VAX, HP, Tandem, Pyramid, NCR) IBM mainframes (MVS, VM) RS/6000 (£10000 3 for price of 1 licence academic discount - other licensing arrangements available)	Rules Objects Knowledge islands Currently lacks facility for uncertainty?	Forward Backward Semantic gates Context links	C COBOL FORTRAN Oracle, Rdb, INGRES, ONTOS, Lotus, SQL/DS, Sybase, Informix Win3 Toolbook Hypercard, Supercard, Plus, Guide Excel CAD & CIM (eg Autocad) Dynacomm	Specifically for embedding into programs! Graphical environment (consistent across all platforms) Library modules Integrates both flat files and rdbs Debugging can be carried out using scripts Open Interface - toolkit for building interface : generates C code	Poor documentation (feb.1990) - maybe improved by now

ORION (Machine Reasoning 0276-71141)	IBM RS/6000 £2500 - but discount maybe available	Equations!		Written in Pascal	X-Windows.	No sales yet in this country. An Australian product. Some significant bugs. Poor documentation and support. Useful equation solver allowing implicit equation format
ProKappa (IntelliCorp UK)	Sun SPARC station Sun3 HP RS/6000 (Beta) OS/2 (Beta) MVS (Beta) Runtime licence For X-windows: 24 MB RAM recommended	Objects Rules	Pattern-matching Variables For+Back Backtracking Multiple and selective inheritance	Written in C Free use of C	Embeddable Program in C and/or ProTalk (own lang.) Interactive env. Incremental compiler and linker Debugger (inc. "Probes" attached to anything) Substrate: symbolic C prog. (new data-types, type-checking, memory man., prog. error hand.)	RS/6000 version available Sept 1991. Other platforms will include those required by the ACE consortium. ProKappa on Suns is used in UCL Statistical Science - contact Mark Leaning x3638. New Scheduler and Configurator booster modules (class and function libraries which may be used directly or as example source code) available. **Suitable as the basis of the RISC system.**
SIMUNC		Rules and uncertainty				Research tool.
TIRS (IBM)	Only IBM OS/2 CMS TSO VMS CICS				Environment + incremental compiler Debugging tools	Seen at an IBM seminar - environment seems very powerful.
Trellis (Digital)	Only DEC					
VAX OPS5	VAXstation	Production rules				See [Mettrey].

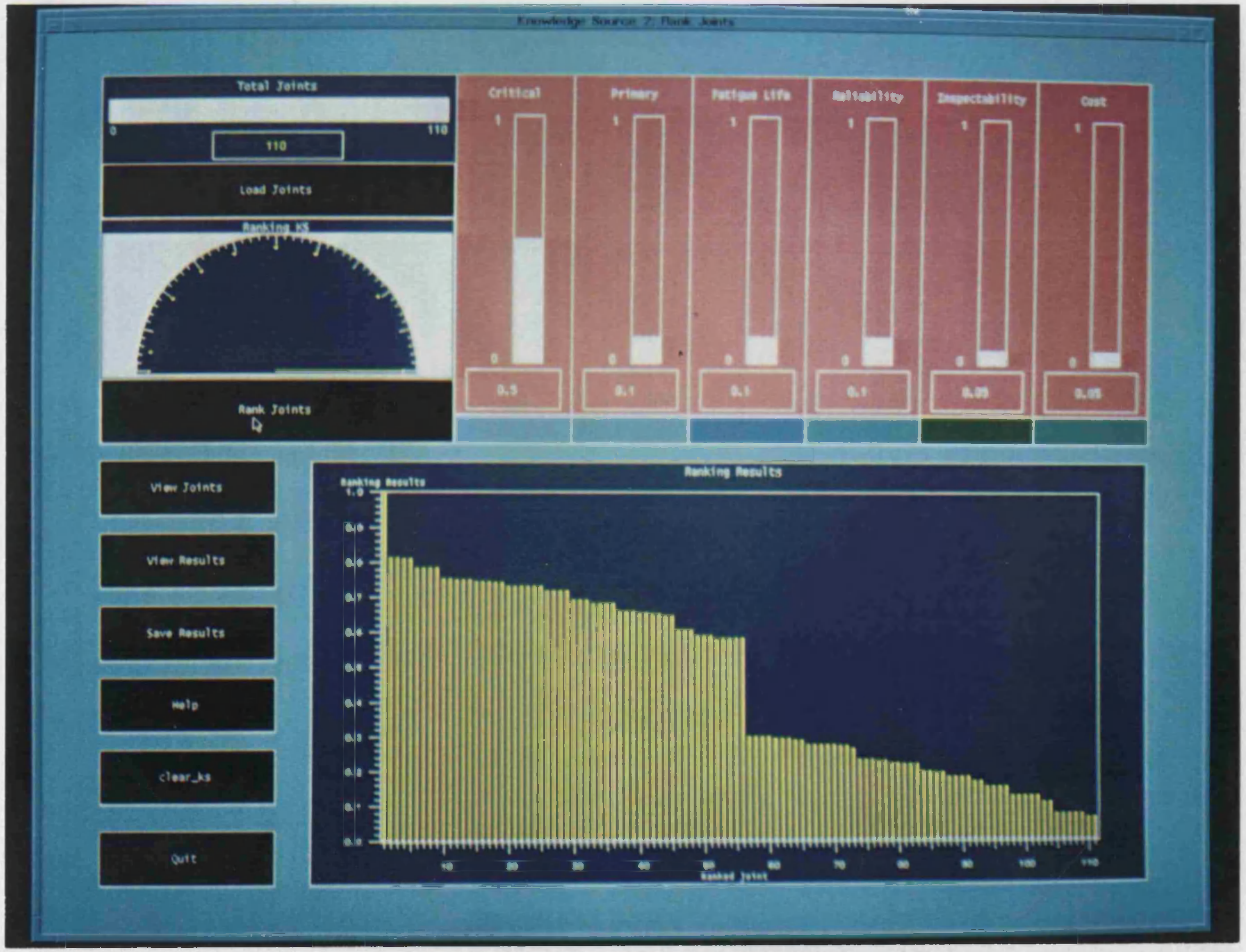


Figure 5.22 Ranking module in the RISC Demonstrator

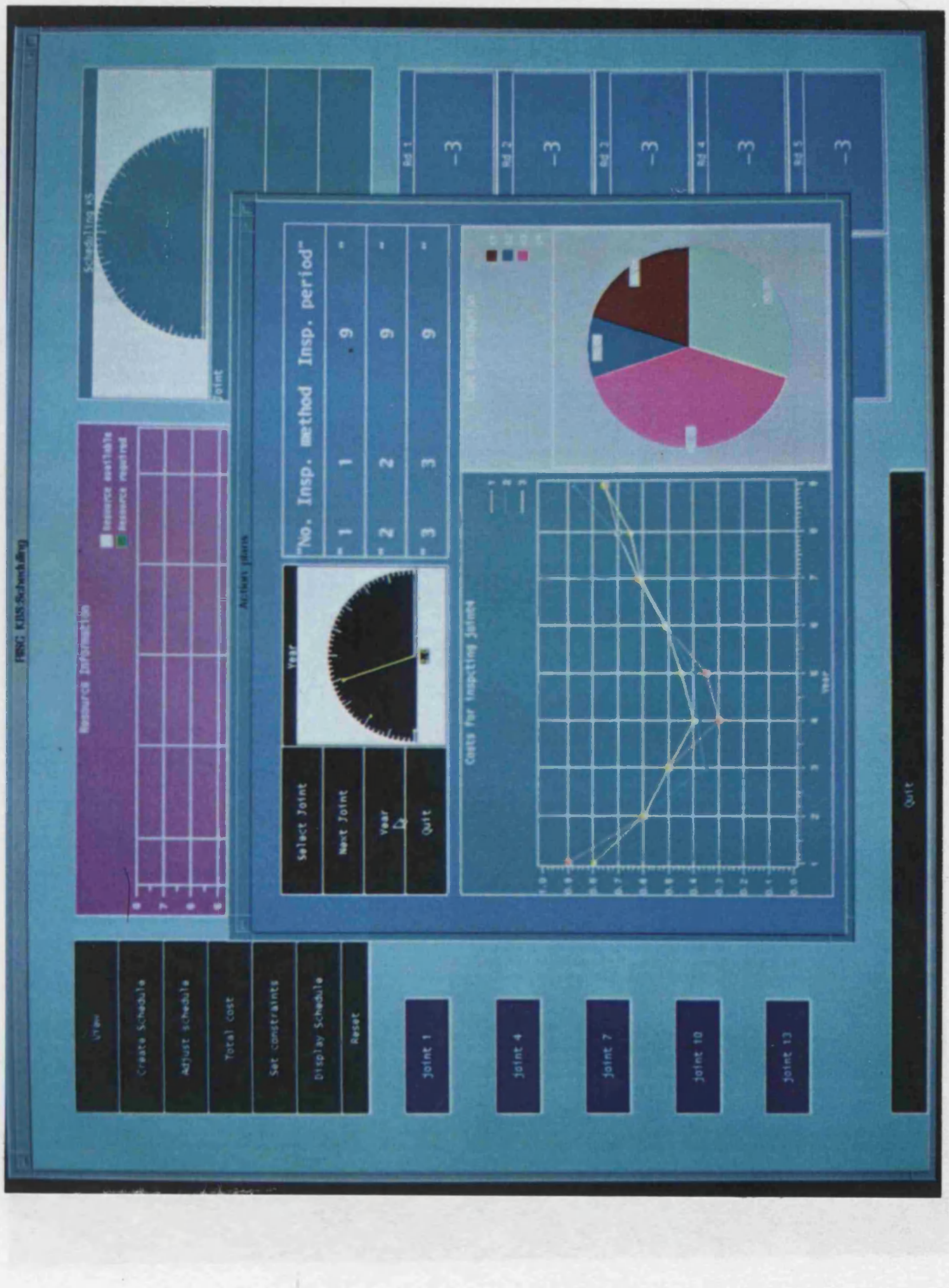


Figure 5.23 Display of RISCREL cost evaluation results

Figure 5.24 An initial schedule with the RISC Demonstrator

Figure 5.25 A usable and rational schedule after resource allocation

6 A CASE STUDY

Earlier chapters explained the basis for the RISC methodology, gave specifications for the RISC System and described the development of a prototype RISC System. In this chapter, a case study is given to illustrate the RISC methodology. The use of RISCREL is explained and details of the joints and cases investigated are given. Values for the random variables and the parameters are specified for the example joints, with explanations as to how this data was derived or with references to other work using similar values. An important input to the RISC analysis is past inspection information and this requires some understanding of how to make use of the appropriate information on inspection reliability. The problems that arise from this are touched upon here.

Section 6.2 describes the data required and gives example values. Section 6.3 explains how the RISCREL analysis is executed with the data gathered in the earlier section. Finally, the problems of interpretation of the results still remain. The results from RISCREL were interpreted to give a reasonable set of possible maintenance actions for each joint. The interpreted results must then be combined to form a rational global solution for the structure which takes into account constraints which may not have been included (implicitly or explicitly) into the analysis. Section 6.4 shows how the scheduling algorithms are applied to give a complete schedule for the structure.

6.1 THE EXAMPLE STRUCTURE

This section explains some of the problems and considerations to be made in collecting data required for executing RISCREL. The case study data and further examples of realistic data collected are also given.

Most of the information on the appropriate distributions and data for the second or higher order moments, that is the standard deviation and other measures of distribution, was collected from the literature and studies carried out by others during the RISC project. In practice, it was found that apart from some very specific information, such as monitored environmental data and the history of inspections and repairs carried out on the structure under consideration, operators do not have all the required statistical data. Some of this data will need to be found, as is described in this section, either by using independent studies or by approximation.

Some of the example data on the representative or characteristic values, related to the first moment, on random variables was obtained from actual structures. The structure considered in the case study, however, was a fictitious structure with an assumed service life of 25 years. The service life has no effect on the reliability analysis of a joint, since reliability is only affected by past events.

On the other hand, it does affect the costs evaluation of any planned maintenance action, since the economics depend on the future of the structure.

6.1.1 Notes on the use of RISCREL

Input variables for RISCREL can be one of three types:

- control parameters for the structural reliability analysis
- random variables which represent data for which there are uncertainties
- parameters used to input data which is inherently non-random or which signals a fatigue fracture mechanics analysis choice

Apart from a few exceptions, the control parameter values are not varied; the exceptions are explained in Section 6.3.2. The parameter and variables are given in Table 6.20 at the end of this chapter. This table indicates that these sets of data are stored as the values of the attributes of joint type objects. Unless specified otherwise, where no units are given in tables or data below, an appropriate S.I. unit should be assumed. Time variables, however, are in years and angles are given in degrees.

In describing the case study data, only the random variables and the most relevant parameters are given. For the random variables, RISCREL allows four moments or distribution parameters to be given to model each variable, but for the purposes of this exercise, the distributions selected for the random variables only required two moments.

As an example of a RISCREL parameter, L_TIME represents the service life of the structure. For this case study L_TIME was set to 25 years. Other parameters include the choice of stress intensity factor method, loading, and so on.

6.2 MODELLING CASES FOR RISCREL

The data was obtained from various sources. Several similar past case studies have been carried out based on IMREL, as given in, for instance Goyet et al (1994) and other probabilistic analyses, such as Faber et al (1992) and thus the data for these is re-used and discussed. Other data was derived from information from past work carried out in the NDE Centre at UCL on inspection reliability and fatigue fracture mechanics.

6.2.1 Geometry

The geometric information required by RISCREL includes values for random variables modelling:

- member thickness, that is the thickness of the tubular member, which can be either the chord or the brace through which the crack is growing
- the assumed initial crack size, before the results of any inspection and both in terms of depth and length
- the weld geometry, that is, weld toe radius, angle and leg length

These may be modified by any repairs carried out. Hence for each of the geometric random variables, there are corresponding post-weld repair and post-grind repair random variables. The effect of repair on the geometric variables is considered in more detail in Section 6.2.2.

In this section, the values of these variables are given for each example joint. The complete list of random variables is as follows:

T_CHORD	Thickness of the tubular member of interest, usually the chord (units in metres). This is set to the brace thickness if it is known that a crack is growing through the brace.
A0	Initial crack depth (m)
C0	Initial crack length (m)
T_R1	Chord thickness after weld repair (m)
A0_R1	Initial crack depth after weld repair (m)
C0_R1	Initial crack length after weld repair (m)
T_R2	Chord thickness after grind repair (m)
A0_R2	Initial crack depth after grind repair (m)
C0_R2	Initial crack length after grind repair (m)
W_T_R	Weld Toe Radius (m)
W_LEN	Weld Leg Length (m)
W_ANG	Weld Angle (degrees)
W_T_R_G	Weld Toe Radius after grind repair (m)
W_LEN_G	Weld Leg Length after grind repair (m)
W_ANG_G	Weld Angle after grind repair (degrees)
W_T_R_W	Weld Toe Radius after weld repair (m)
W_LEN_W	Weld Leg Length after weld repair (m)
W_ANG_W	Weld Angle after weld repair (degrees)

In addition to the geometry, the last inspection result is also of significance. This data is introduced by setting Y_INSP, representing the year in which inspection took place, and A_INSP, the depth of the observed crack. Different values for these were set for each joint of interest.

Weld geometry variables, W_T_R, W_LEN and W_ANG, were not given values for most of the example joints, as they were not required for the selected analysis route. In general it was assumed that these values were unchanged after repair.

The initial crack sizes A0 and C0 represent the depth and half-length of the cracks in the component at the start of the life of the structure. Since the fatigue fracture mechanics modelling implemented for RISCREL does not model the initiation phase of the fatigue process, the value given to A0 is of great interest since it will have a direct impact on the predicted fatigue life of the tubular connection and hence on the reliability analysis results.It is assumed in general that initially there is only one microcrack at the hot-spot area of the joint. If no other information is known or has been given, common practice has been to employ an exponential distribution with mean value 0.12mm (Diamantidis et al, 1991) or mean 0.11mm (Kirkemo 1988) as a model for the initial crack depth. The Marine Technology Directorate report suggests initial crack values for flat plates under different loading cases, ranging from 0.12 mm for a plate 16mm thick under membrane tension or 0.22mm, to 0.57mm for a 107.95mm thick plate under bending (MTD, 1989). For a plate of thickness 38.1 mm, the MTD report suggests a suitable initial crack depth of 0.18mm and 0.34mm for loading under tension and bending, respectively. Very recent work has suggested that a more realistic model may have a mean value of 0.38mm (Moan et al, 1997). For the RISC project, a value of 0.15mm was assumed (Dharmavasan et al, 1994b).

Finally, A0 may be updated with the past inspection result. That is, if a crack has been previously detected and not repaired, as in joint J02 below, then A0 may be set to the detected crack size. Thus this variable may be used to represent in some way the past history of the joint.

Several tubular geometries were chosen for the fictitious structure and some are illustrated in Figure 6.1. Values for their geometric parameters are given in Table 6.1. Where values for the post-repair variables are not specified, it can be assumed that they are the same as the pre-repair values.

Table 6.1 Tubular joint geometric data

Variable	Dist.	1st Moment	2nd Moment	Remarks
Joint A				From RISCREL Manual
T_CHORD	weibull	0.040000	0.005000	
A0	normal	0.000800	0.000100	

Variable	Dist.	1st Moment	2nd Moment	Remarks
C0	normal	0.008000	0.000800	
Joint B				From Goyet et al (1994)
T_CHORD	lognormal	0.038100	0.005720	A0 is based on the results
A0	lognormal	0.006000	0.000500	of inspection in the first
C0	lognormal	0.008000	0.000800	year.
Joint C				Example joint from actual
T_CHORD	lognormal	0.021437	0.000214	structure
A0	lognormal	0.000150	0.000015	Medium sized member with
C0	lognormal	0.000750	0.000075	no cracks
A0_R1	lognormal	0.000100	0.000010	Repairs are expected to
C0_R1	lognormal	0.000500	0.000050	reduce initial crack sizes
A0_R2	lognormal	0.000100	0.000010	
C0_R2	lognormal	0.000500	0.000050	
W_ANG	normal	45	1.000000	
Joint D				Larger members
T_CHORD	lognormal	0.038100	0.005720	No cracks
A0	lognormal	0.000150	0.000015	
Joint E				
T_CHORD	weibull	0.040000	0.005000	
A0	normal	0.000800	0.000100	
T_R1	weibull	0.038000	0.005000	
A0_R2	normal	0.001000	0.000500	
Joint F				
T_CHORD	lognormal	0.038100	0.00571	
A0	normal	0.000600	0.000060	
A0_R1	normal	0.001500	0.000150	
T_R2	lognormal	0.036100	0.00541	
Joint G				
T_CHORD	lognormal	0.038100	0.00571	
A0	lognormal	0.000150	0.000015	
A0_R1	lognormal	0.001000	0.000100	

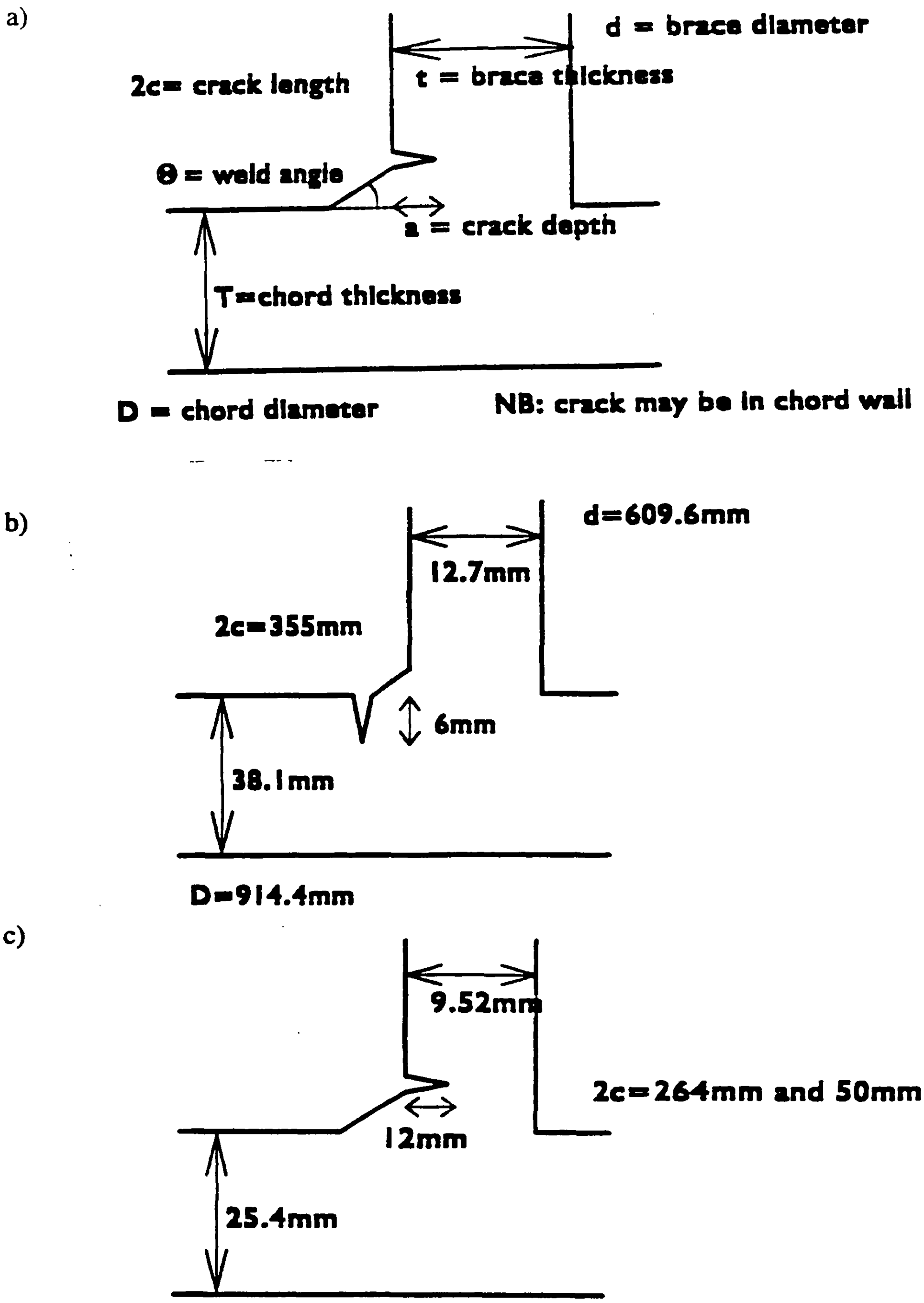


Figure 6.1 Case study tubular connections (a) General notation (b) from RISC Final report (1994) (c) in Goyet et al (1994)

6.2.2 Modelling of Repair

Two methods of repair may be considered at one time in RISCREL. The modelling allows the possibility of grinding being carried out if a small crack is found, or welding if the crack is considered to be too large to be repaired by grinding alone.

RISCREL allows changes due to repair actions to the geometry of the joint, that is the thickness of the member, initial crack sizes and weld toe radius, leg length, and angle. For instance, in the case of welding it is expected that the wall thickness be the same as at the start of the life of the joint, but with added uncertainties as to the initial crack sizes, since underwater welding is necessarily of lower quality than welding in fabrication. Additionally, after grinding repairs, the wall thickness is reduced by a few millimetres, although the uncertainties in the initial cracks will be as at the start of the life of the joint. The chord thickness after weld and grind repair is modelled by the random variables T_R1, T_R2 respectively. Similarly, the initial crack size, that is depth and width, after weld and grind repair are modelled by the random variables A0_R1, A0_R2, C0_R1 and C0_R2 respectively. The toe radius, leg length, and angle of the weld after repair are modelled by W_T_R_G, W_LEN_G, W_ANG_G, W_T_R_W, W_LEN_W, and W_ANG_W, where the suffixes G and W represent grind repair, or R2, and weld repair, or R1, respectively. These variables were assumed for the purposes of this case study to have the same values as for the corresponding pre-repair variables.

The costs of the two repair methods are introduced by way of the parameters C_Rep1, corresponding to welding, and C_Rep2, corresponding to grinding. The cost of welding is usually significantly higher than for grinding only, since grinding is usually carried out by the inspecting diver if any crack is detected. If welding is required, this would require a new team of divers with the appropriate equipment.

Table 6.2 Default data for modelling of repairs

RV	Distribution	1st Parameter	2nd Parameter
T_R1	as T_CHORD	as T_CHORD	10%T_R1
A0_R1	as A0	as A0	10%A0_R1
C0_R1	as C0	as C0	10%C0_R1
T_R2	as T_CHORD	T_CHORD-0.002m	10%T_R2
A0_R2	as A0	as A0	10%A0_R2
C0_R2	as C0	as C0	10%C0_R2
W_T_R_G	as W_T_R	as W_T_R	as W_T_R
W_LEN_G	as W_LEN	as W_LEN	as W_LEN
W_ANG_G	as W_ANG	as W_ANG	as W_ANG
W_T_R_W	as W_T_R	as W_T_R	as W_T_R
W_LEN_W	as W_LEN	as W_LEN	as W_LEN
W_ANG_W	as W_ANG	as W_ANG	as W_ANG

Finally, the repair criterion parameter R_Rep is used to indicate when each repair is carried out. For instance, if R_Rep is set at 0.005m, then this means that grinding is carried out if a crack is detected and has depth less than 0.006m, otherwise if the crack depth is greater than 0.006m then welding is employed. This example repair criterion value is appropriate since it is expected that if more than 6mm has been ground away, then welding is more likely to take place. In Faber et al, the repair criterion was set to 0.01m, but this is higher than may be expected in practice. A more appropriate value can be set by considering the original thickness of the joint: if the total material already ground away during previous repairs is 0.002m then R_Rep would be set at 0.004m, that is by deducting 0.002m from the total maximum grinding depth permitted, say 0.006m.

Repair may also affect the material properties. This is discussed in the following section.

6.2.3 Material Properties

Since the failure of a tubular joint is modelled using Paris' Law as described in Section 2.3, the data required to define the corrosion fatigue properties of a material is

- the Paris constant C
- the Paris exponent m

As described in Section 2.3, it is important to enable a multi-segment Paris Law relationship to be described to model crack growth in a corrosive environment. In RISCREL, up to five values for C and for m are allowed, each pair of values corresponding to each segment of the Paris Law curve. Paris_C1, Paris_C2, Paris_C3, Paris_C4, and Paris_C5 are random variables, whilst the corresponding Par_M1, Par_M2, Par_M3, Par_M4, and Par_M5 are parameters. The number of segments to be used is given by the parameter D_Seg. In order to make use of the different values of C and m, the loading must be modelled as a stress range probability distribution and this is explained in Section 6.2.4.

Welding, that is, repair type R2, may change the material properties, in which case the random variables Paris_1, Paris_2, Paris_3, Paris_4, and Paris_5 may be set to different values to reflect this. In general, however, these variables are set to the same values as for before welding. The values for the above are stored in a database.

Actual materials considered were

- E46 steel with modulus of elasticity 210000MPa and Poisson coefficient = 0.3 (Goyet et al, 1994)
- BS4360-50D steel as used in an example in Kam (1989a)

The remaining example values are for materials not identified in the original reference. The values used in the case study are in the table below. The Paris constant C after weld repair can be assumed to be the same, unless otherwise given, and the default number of segments is 1.

Table 6.3 Material properties

Material	C dist	C mean	C s.d.	Par_M	Remarks
E36	LN	5.00E-12	5.00E-12	3.320	From Goyet et al, (1994)
BS4360-50D	LN	4.50E-12	1.35E-11	3.30	Kam (1989a)
M1	LN	1.37E-08	0.95E-08	3.019	Material in use in Mexico
M2	N	5.00E-12	5.00E-12	3.100	From RISCREL Manual
M3	LN	1.13E-10	2.825E-11	2.00	From Faber at al (1992)
M4	LN	4.50E-12	3.10E-12	3.100	From RISCREL Manual

The Marine Technology Directorate report on underwater inspection procedures also suggests values for C and m for different conditions (MTD, 1989). For example for steel tubulars in seawater and under cathodic protection, m is given a value as high as 5.96 and C a value of 3.5×10^{-12} in MN, m units.

6.2.4 Loading

Two methods of defining the loads are provided. The first method describes the load as one Weighted Average Stress Range (WASR) random variable; the second method allows the modelling of a stress range probability distribution. If the crack is being modelled in 2 dimensions or if corrosion effects are to be taken into account, that is a multi-segment Paris Law is to be used requiring several stress values, then the WASR load model should not be used, instead the stress range probability distribution model is required.

The Weighted Average Stress Range is a single stress value which gives the equivalent damage to that produced to all the random stress cycles applied to the structure: the total crack growth produced by the WASR is equal to the total crack growth produced by all the different stress ranges. A value and distribution for the random variable WASR is required. This variable represents the so-called hot spot stress for the joint, which, as described in Section 2.3.3.1., is the product of the nominal stress and the stress concentration factor. Reasonable values for the mean of WASR lie in the range 30-40MPa, which corresponds to a design S-N curve for tubular welded structures with a required 2.3% or less probability of failure for a fatigue life limit of 10^5 cycles (Maddox, 1991; Dharmavasan, 1995).

The stress range probability distribution (SRPD) model allows a more accurate corrosion fatigue crack growth model. As described in Section 2.3.3.1, a two-parameter Weibull distribution is used to model the applied stresses and thus two variables Weibull_A and Weibull_B have to be set. The loading analysis program, ULDAN, will provide the Weibull parameters for the stress load distribution and for this, the spectral moments are required. As the case study made use mainly of WASR method of load modelling, ULDAN was not executed during the case study. Two cases included the use of the Weibull model as a comparison, and these Weibull parameters were taken from past examples. In addition, the number of loading cycles per year, represented by the parameter Cyc_Y, is required. Realistic values for Cyc_Y range from 10 thousand to 10 million (Faber, 1992; Kam, 1989b).

- **WASR**

To select this method, the parameter L_Meth is set to 0. The WASR models long-term loading on the weld of the joint. The mean value for WASR represents the *hot spot stress* given by:

$$\text{WASR} = \text{Nominal Stress} \times \text{Stress Concentration Factor} \qquad (6.1)$$

Table 6.4 Loading modelled as a WASR

Distribution	1st Moment	2nd Moment	Remarks
normal	18.0	1.50	From RISCREL Manual
normal	30.0	4.00	From RISCREL Manual
normal	20.0	6.000	From RISCREL Manual
lognormal	61.6	18.48	From Goyet et al (1994)
lognormal	10.0	3.00	Mean value from Goyet et al (1994), s.d. from 18.48/61.6=0.3

Realistic mean values for the WASR lie in the range 30-40MPa and a good estimate for the standard deviation is that which gives a coefficient of variation of approximately 0.3. As the WASR cannot take on negative values, a particularly suitable model is the lognormal distribution, although, for sufficiently large mean values and small coefficients of variation, the normal distribution is a reasonable assumption.

- **Weibull Method**

For this method, L_Meth is set to 3. As the two variables Weibull A and Weibull B have to be set together, in the following table each pair defines the load.

Table 6.5 Loading modelled as a Weibull distribution

Weibull P.	Dist.	1st Moment	2nd Moment	Remarks
Weibull_A	normal	1.5300	0.3220	From RISCREL Manual
Weibull_B	normal	1.0190	0.3220	

6.2.5 Fatigue Crack Growth Modelling

For the fatigue fracture mechanics element of the analysis, it is necessary to consider two aspects:

- the stress intensity factor (SIF) solution
- modelling of the crack-growth in one direction only (depth) or in both the depth and length directions

For the purposes of the case study, it was decided to concentrate on 1-dimensional modelling of cracks. In this section then, only the choice of SIF solutions are discussed in detail.

6.2.5.1 <u>SIF Method</u>

The choice of SIF equation is made by setting the parameter SIF_M to

1 for AVerage Stress SIF method (AVS)

2 for the Two Phase Method (TPM)

3 for the Holbrook-Dover solution

4 for the Newman-Raju solution

Both AVS and TPM are empirical SIF relations for tubular joints, which are based on experimental crack shape developments in tubular joints. Newman-Raju and the Holdbrook-Dover solutions, in contrast, are flat plate models. Hence, this case study has concentrated on the use of AVS and TPM.

- **AVS**

For the AVerage Stress SIF method, it is necessary to set parameter values for the hot spot stress concentration factor HS_SCF, and the average SCF AV_SCF. The valid ranges of values are given by 2.66< HS_SCF <9.1 and 1.55< AV_SCF <6.35. The ratio HS_SCF/Av_SCF is the actual measure used in the AVS equation, and this is normally close to 1.0. For the purposes of this case study, values of 1.0 have been generally assumed for both AV_SCF and HS_SCF.

Table 6.6 Data for the AVS SIF solution

	HS_SCF	AV_SCF	Remarks
Default	1.00	1.00	Used in the RISCREL Manual
	3.14	6.28	From RISC Final Report

- **TPM**

For the Two Phase Method, it is necessary to set parameter values for the hot spot stress concentration factor HS_SCF, the average SCF AV_SCF and a geometric parameter Geo_P. Geo_P corresponds to the tubular joint geometric ratio $\beta = d/D$, where d is the diameter of the brace and D is the diameter of the chord. Thus Geo_P is always less than 1 in value. The valid ranges of values are 0.21< Geo_P<1.0. The default value of Geo_P was taken to be 0.71 when the full geometry of the joint was unknown. The same validity ranges as for the AVS method for HS_SCF and AV_SCF values apply. As TPM, like AVS, makes use of the ratio HS_SCF/AV_SCF only, the same comments on default values apply. When the TPM option is selected, the analysis is sensitive to the exact loading input values and does not converge for high loading values. Thus, this option was not used for the study.

6.2.5.2 <u>2-Dimensional Crack Modelling</u>

As an alternative to the 1-dimensional crack growth model, which relies on direct information on depth, a 2-dimensional model of the crack growth was included in RISCREL. This model assumes a semi-elliptical crack shape development, which models fairly well the case under tension or bending, of depth a and length $2c$. The crack growth is calculated by carrying out integration in two dimensions: along the length c and the depth a. Hence two crack growth laws are employed and are specified by the parameters D_G_L, the depth growth law, and L_G_L, the length growth law. Default values for these are 1.00. In addition, the parameter representing the Weibull bias factor Bias_F is set. Kirkemo (1988) reported a typical value for the bias factor mean to be 0.70. In the RISC project, a bias factor value of 0.71 was employed. The geometry function is here represented by the parameter Mk_factor. This is set to a value which indicates the 2-D model to be used. For these, values for other factors, such as the half_plate_width, are required. This option was not used in the case study.

6.2.6 Modelling of Inspection

One of the fundamental requirements is to be able to model accurately the effect of inspection. As already mentioned in Section 6.2.1, RISCREL allows one previous inspection result to be taken into account by setting the parameters A_INSP, the crack depth at the last inspection, and Y_INSP, the year of this last inspection. An error condition occurs when the inspected crack size is smaller than at the beginning of the life of the structure: therefore A_INSP must be greater than A0. Additionally, RISCREL allows modelling of inspection reliability by employing random variables:

POD	Probability of detection of the inspection technique employed in year Y_INSP.
Epsilon	The sensitivity of the technique employed, which is also known as the probability of sizing (POS).
POD_P	The planned inspection technique POD
Epsilon_P	The planned inspection technique POS

To be able to measure the POD and POS requires the production of extensive data on detection of many cracks of different sizes. This introduces the problem of ensuring that sufficient and representative samples exist. Furthermore, the tendency of a technique to detect spurious cracks should also be considered. At the present moment, the false-call rate is as yet not incorporated into the RISC system. This is acceptable as the current assumption is that detected defects are immediately checked, which would have the effect of reducing the false call rate. In a practical situation, once a crack has been detected, often a separate inspection technique may be employed to size the crack. In effect two inspection techniques may be used, one for detection, the second for sizing, and the input POD and POS values for both the performed and planned inspections need not correspond to the same technique. Some information was available from previous work and used in the RISCREL Manual (see Dharmavasan et al, 1994b) on the following representative inspection techniques

- above water magnetic particle inspection (AWMPI)
- underwater magnetic particle inspection (UWMPI)
- above water eddy current inspection (AWEC)
- underwater eddy current inspection (UWEC)
- above water close visual inspection (AWCV)
- underwater close visual inspection (UWCV)

In addition, up-to-date data was collected for the alternating current field measurement (ACFM), the Thorburn electromagnetic detection system (EMD III) and the Hocking AV100 (AV100) inspection techniques. Further work on inspection reliability has been carried out within the InterCalibration of Offshore NDT (ICON) project at University College London and some results have been used for this case study (Rudlin & Dover, 1996; Rudlin, 1996).

6.2.6.1 POD and POS

Work has been carried out in modelling POD for these techniques using the exponential, logit model and Weibull distributions and the results of this were modified to be able to be used in the RISCREL program (Ximenes & Mansour, 1991; Rudlin & Wolstenholme, 1992). Fictional data was also included in the study to represent levels of expected sensitivities for different quality techniques. The exponential distribution was assumed for all techniques, as this corresponds to much of the evidence. For the exponential distribution, the relationship between the mean and the standard deviation is given by

$$\text{mean} = \gamma + 1/\lambda \quad \text{(a)} \qquad \text{s.d.} = 1/\lambda \quad \text{(b)} \qquad (6.2)$$

where λ and γ are the first parameter and the second parameter, respectively. Results for the ACFM technique indicate that the detection capacity of ACFM is very similar to that of MPI. The following tables shows the data used in the case study and the corresponding mean values (correct to 4 decimal places) for the depth of the undetected crack.

Table 6.7 Inspection POD data

Technique	Distribution	1st Parameter	2nd Parameter	Mean(m)
AWACFM	exponential	0.000010	2000.00	0.0005
UWACFM	exponential	0.000010	1000.00	0.0010
AWMPI	exponential	0.000010	1000.00	0.0010
UWMPI	exponential	0.000010	770.00	0.0013
AWEC	exponential	0.000010	667.00	0.0015
UWEC	exponential	0.000010	500.00	0.0020
AWCV	exponential	0.000010	167.00	0.0060
UWCV	exponential	0.000010	76.90	0.0130

The values for the standard deviation of each can be easily deduced and are conservative. The mean values of the undetectable crack size seem to correspond to the intuitive estimates given by inspection experts informally. Additionally, the parameters for the POD using the EC technique underwater (UWEC) correspond well with the results reported by Moan et al (1997) of a mean value of 1.95mm.

Information on the sizing accuracy or POS was not immediately available for all inspection techniques. Some limited data from the ICON project has been published for the ACFM technique indicating that a normal distribution with mean 0.0 and standard deviation 0.001 is a conservative model (Dover et al, 1994). The data in the table below was used.

Table 6.8 Inspection POS data

Technique	Distribution	Mean	S.D. (m)
ACFM	normal	0.00	4.00E-04
AWMPI	normal	0.00	4.00E-04
UWMPI	normal	0.00	8.00E-04
AWEC	normal	0.00	1.20E-03
UWEC	normal	0.00	1.60E-03
UWCV	normal	0.00	2.50E-03

6.2.7 Costs Data

Costs are required for failure, each repair method, and for inspection. The range of cost values used in the case study are given in Table 6.9.

Failure costs are determined by considering the redundancy or criticality of the component combined with the costs associated with stopping production to allow emergency repair work. If systems reliability were implemented then of course redundancy would not be taken into account at this point. Costs of welding are always high in comparison to grinding repairs. The latter can be assumed to be of the same order as the costs associated with inspection. Inspection costs vary according to the technique employed and the location of the component. Beyond a certain depth, saturation diving is required and this will increase the costs considerably. If the inspection rate is one node per day, then the cost per node would be the same as the diving costs per day that is £30,000 (MTD, 1989).

A fuller study may be required which identifies the separate costs more accurately, such as what part of the inspection cost is due to the location as opposed to the technique employed. Inspection costs can be modified if it is be assumed that in one diving session more than one component can be inspected, such as in the situation where groups of components are physically near to one another. This modification can be difficult to assume at this point since it makes the assumption that several neighbouring joints will be inspected at the same time.

The effects of inflation may be considered by setting the parameters R_RATE, representing the real rate of interest, and Y_CAPI, for the year of capitalisation. For the case study, this was not considered.

Table 6.9 Case study costs

Event	Cost (£)	Remarks
Failure of critical node	100,000,000-10,000,000	Estimate provided by reliability and NDI experts (MHF, JR)
Failure of a non-critical but not redundant node	10,000,000-1,000,000	Estimate provided by NDI expert (JR)
Failure of a minor node	1,000,000-100,000	
Weld Repair	50,000-100,000	Lower values was used by MHF, higher value estimated by JR
Grinding	5,000-25,000	Lower values are for easily accessed nodes
Inspection costs	3,000-50,000	
Close visual inspection	3,000-25,000	The lower values correspond to above water inspections, the highest value assumes deeper location.
MPI inspection	3,000-50,000	This includes cleaning costs.
EC techniques	4,000-50,000	Lower estimate given by expert for EMD III and for the AV100 ACPD technique.
ACFM	5,000-50,000	Revised estimates

6.3 THE RISC ANALYSIS

The RISC analysis includes ULDAN and RISCREL. The load analysis program, ULDAN, will produce the two parameters of a Weibull curve modelling the stress range distribution. As it was not used for the case study, it is not described further in this chapter. The issues related to RISCREL are covered in greater detail.

6.3.1 The Tubular Components and Maintenance Plans

The actual structure used as a case study is a fictitious one taken from the MTD report (1989). The structure is shown in Figure 6.2. In this structure, several critical joints can be identified. There are joints which share geometry, but may be under different loads, or indeed may have different defects. This a fairly typical example of the type of structure to which the RISC methodology may be applied.

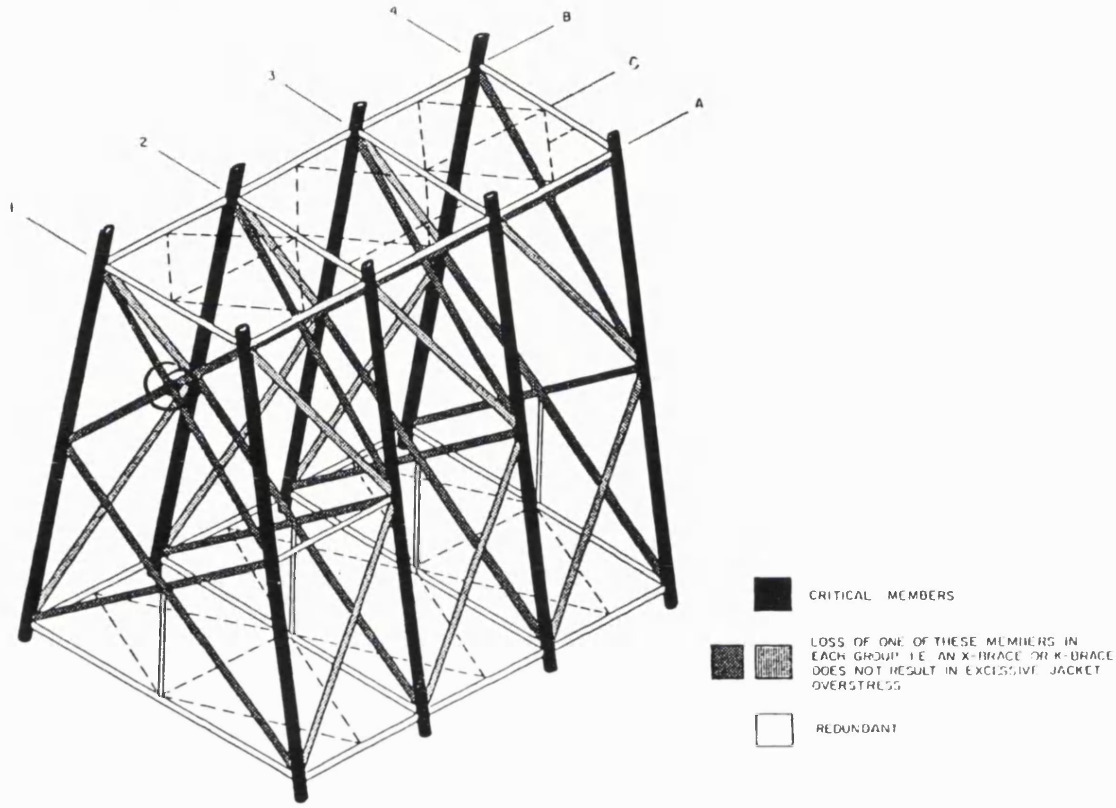


Figure 6.2 Case study structure

The nodes on the structure are identified by noting the level (1 at the top, 5 at the bottom). Then on the horizontal plane: A denotes nodes on the side closest to the observer, B denotes nodes those furthest away and C for nodes in the middle of the structure, and 1 to 4 from front to back of the structure. Thus the node ringed in the diagram is identified by 2C1. The largest and critical joints are part of nodes 1A1 to 4, 1B1 to 4, 3A1 to 4, 3B1 to 4, 5A1 to 4, and 5B1 to 4. Nodes 2C1 to 4 and 4C1 to 4 will contain only non-critical but not minor joints. Joints at nodes 5C1 to 4 may be safely ignored as they are very minor nodes. Of the primary and non-redundant nodes, some are very similar, not just in terms of their geometry but also in terms of the loads applied.

The joints are identified by considering the brace member. A member can be identified by its two end nodes. So for a node, giving the node at the other end of the member identifies the weld or joint of interest and adding "c" or "b" differentiates between a crack through the chord wall or the brace wall. For example, at node 2C1 there are two welds joining members 1A1-2C1 and 2C1-3B1 to the chord 1B1-3A1. These welds can be named 1A1 and 3B1. The possible joint cases which can be considered for analysis are 1A1c, 1A1b, 3B1c and 3B1b.

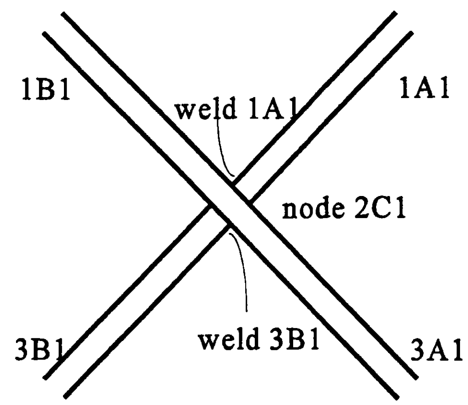


Figure 6.3 Example joint identifiers

There are a total of 64 joints to consider for this structure (ignoring all redundant members), in addition to which, cracks may grow in either the chord or brace wall for each joint, making a total of 128 reliability analyses. Because of this and to reflect what would occur during the use of RISC, joints with the similar geometry and service history are not analysed individually.

The geometries identified in Section 6.2 were augmented by considering joints which had the same initial geometry, but different past inspection results. In total, over fifteen joints were analysed. These corresponded to some of the nodes and their correspondence is shown in Table 6.10, which also indicates which nodes were considered to be similar and hence each joint and maintenance case was analysed once. To make the table easier to read, joint case 6 is for a small member with low loading; cases 2, 3, 4, 5, 7 and 8 have all the same medium sized geometric dimensions and are for the same load or similar load case; case 15 has the same medium geometry, but higher loads; cases 9, 10, 11 are for large members and low loading; 12 and 13 for medium loads on a large member and 1 and 14 are for high loads and a large member. Case 6 corresponds to the minor nodes not considered in the tables below.

Approximate fatigue lives at design of the joints were known and only eight had low fatigue lives: the welds on nodes 4C1-2 had fatigue lives between 10 and 30 years, nodes 4C3 and 4, between 30 and 100 years. For the purposes of the study, some of the joints are assumed to have been inspected in the past and some cracks found. The ranking would be carried out based on the criticality of the members making up the joint, the design fatigue life, and the results of past inspections.

For some joints, more than one maintenance plan was considered. Approximately fifty cost evaluation analyses were executed to test the case study and sample cases are shown below.

Table 6.10 Nodes and joints for the example structure

Node	Similar Nodes and Remarks	Joint IDs	Joint Case
1A1	1A2-3,1B1-3	3B1b and 3A2b	3
		3B1c and 3A2c	9
1B2	Crack detected in year 5	3A2c	10
1A3	Crack detected in year 4	3B3b	4
1A4	1B4	3B4b	3
		3B4c	9
1B4	No previous inspection	3B4b	5
2C1	2C2-3	1A1b and 3B1b	2
		1A1c and 3B1c	2
3A1	3B1	1B1b and 4C1b	5
	No inspection at all	1B1c and 4C1c	9
3B1	Two cracks found in year 5 and	5A1b	15
	3 respectively	3C1c	11
3A2	3A3-4,3B2-4	2A2b and 1A1b	3
		4C2b and 5A1b	7
		2A2c, 1A1c, 4C2c, 5A1c	1
3A4	Crack found in brace	1B4b	8
3B3	Crack found in chord	1B2c	14
4C1	4C2-4	3A1b and 5B1b	2
		3A1c and 5B1c	2
5A1	5A2-3,5B1-3	3B1b and 3A2b	3
		3B1c and 3A2c	1
5A4	5B4	3B4b	3
		3B4c	1

Table 6.11 Sample analysis cases

Joint Case 1	Geometry A and material M2
Loading and FFM	WASR = LN(30.0,4.0), 6,000,000 cycles/year
Performed inspection	In year 1, employed AW MPI and no crack detected.
Costs	Failure £1000000, grinding £25000, welding £500000

IRM Plan 1	AW MPI at cost of £15000, weld when >0.01m
IRM Plan 2	As above, weld when > 0.005 m
IRM Plan 3	ACFM, £25000, weld when > 0.005 m
Joint Case 2	Geometry F, material=M1
Loading and FFM	SRPD Weibull(1.53,1.019), cycles 20,000
Performed inspection	Crack depth 0.023m detected in year 4, with ACFM
Costs	Failure £100000, grinding £25000, welding £50000
IRM Plan 1	£15000 UWMPI, weld when > 0.005m
IRM Plan 2	£15000 UWEC, weld when > 0.005m
IRM Plan 3	£10000 UWCV weld when > 0.005m
Joint Case 6	Geometry C, material M1
Loading and FFM	SRPD Weibull(1.53,1.019), cycles 20000
Performed inspection	In year 4, AW MPI employed and no crack detected.
Costs	Failure £1000000, welding £20000, grinding £60000
IRM Plan 1	AW MPI cost £15000, weld when > 0.005 m
IRM Plan 2	AW EC cost £30000, weld when > 0.005 m
Joint Case 8	Geometry G, material M1
Loading and FFM	SRPD Weibull(1.53,1.019), 20000 cycles per year
Performed inspection	Crack of depth 0.01m was detected in year 4, withMPI
Costs	Failure £1000000, grinding £20000, welding £50000
IRM Plan 1	ACFM £35000, weld when > 0.006
IRM Plan 2	UWMPI at £25000, weld when > 0.006
Joint Case 9	Geometry E, material M2 .
Loading and FFM	WASR N(18.00,1.5), and 6000000 cycles per year
Performed inspection	None
Costs	Failure £1000000, grinding £25000, welding £50000
IRM Plan 1	ACFM £35000, weld when > 0.006m
IRM Plan 2	As above, weld when > 0.003m
IRM Plan 3	MPI £25000, weld when > 0.006m
IRM Plan 4	As above, weld when > 0.003m
Joint Case 11	Same as Joint 9 but crack detected
Performed inspection	Crack depth 0.003m in year 3, ACFM

IRM Plan 1	£50000, MPI, weld when >0.005m
IRM Plan 2	£50000, CV, weld when >0.006m
Joint Case 12	Geometry E, material M2
Loading and FFM	WASR(24,2), 6000000 cycles
Performed inspection	Crack depth 0.007m in year 3, 0.04 1000
Costs	Failure £10M, grinding £50000, welding £100000
IRM Plan 1	£50000, MPI weld when > 0.001m
Joint Case 14	Geometry A, material M2
Loading and FFM	WASR (24, 1.5), 60,000,000 cycles
Performed inspection	Crack depth 0.03mm found year 4 inspected with UWEC
Costs	Failure £1000000, grinding £25000, welding £100000
IRM Plan 1	£25000, MPI, weld when > 0.002m
IRM Plan 2	£35000, ACFM, weld when > 0.002m
IRM Plan 3	£25000, MPI, weld when > 0.006m
IRM Plan 2	£35000, ACFM, weld when > 0.006m
Joint Case 15	Geometry D, material BS 4360-50D
Loading and FFM	WASR N(30,4), cycles= 200,000
Performed inspection	Crack detected of depth 0.003m in year 5 with 0.016, 500
Costs	Failure £1000000, grinding £25000, welding 100000
IRM Plan 1	£25000, MPI weld when > 0.006m
IRM Plan 1	£5000 CV, weld when > 0.002m
IRM Plan 1	£35000 ACFM, weld when > 0.005m
Joint Case 16	RISCREL Manual example, geometry A, material M3
Loading and FFM	WASR LN(30, 4) with AVS with hot spot SCF = 1.00 and average SCF = 1.00, 6,000,000 cycles
Performed inspection	Crack depth 0.001m observed in year 1, ACFM

6.3.2 Execution of RISCREL

To execute the analysis module RISCREL, a text input file is created with the data collected and described in the earlier sections of this chapter. RISCREL is run with this input file and several text results files are output by the module. RISCREL may be executed in two modes: updated reliability analysis or cost evaluation of maintenance plans. The mode of running RISCREL is controlled by setting the control parameters as described in Table 6.20 at the end of this chapter. For each joint, at least two analyses were run. The reliability analysis provided the reliability index from which possible latest inspection times was identified. Cost evaluation was carried out for each maintenance plan selected for the joint. The analysis is carried out based on the assumption that the current year is year 5 and an inspection strategy for the next 5 years is the aim.

6.3.2.1 Updated Reliability

To demonstrate the result of different past inspections on the same type of joint in the same environment, one geometry and loading case was chosen for three different joints: the first with no crack detected using MPI, the second with crack found using MPI and the third having a crack detected with ACFM. These results are shown in Figure 6.4 overleaf.

As can be seen, the greater accuracy of the ACFM technique means that the reliability index is updated to a higher value, since there is a reduced uncertainty. The surprising result is that no crack detected does not necessarily mean that the reliability index is updated to a higher value than if a crack has been detected. This could be because the sensitivity of a technique may be such that little information is provided by no crack detection. A detected crack however is likely to be very carefully sized, which leads to more information, or reduced uncertainty.

In the course of this study, it was found that β values are highly affected by uncertainty in material properties, that is in Paris' Law constant C, and to the uncertainty in WASR.

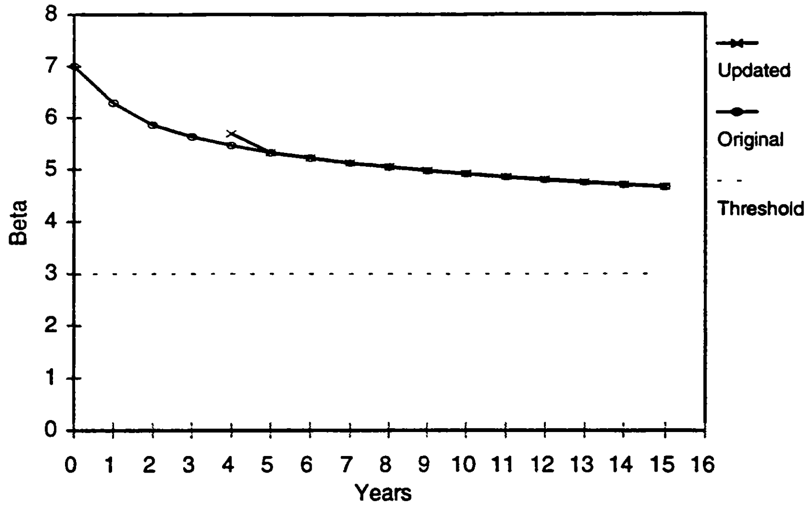
6.3.2.2 Cost Evaluation

The output of cost evaluation includes for each inspection point, the expected costs of failure, repair and inspection, and the probability of failure corresponding to the particular maintenance plan as a function of time to the inspection. Of most interest of course is the expected total cost, that is the sum of the above costs, since the minimum total cost for a joint identifies the optimal maintenance plan for the joint. Examples of results are shown in Figure 6.5.

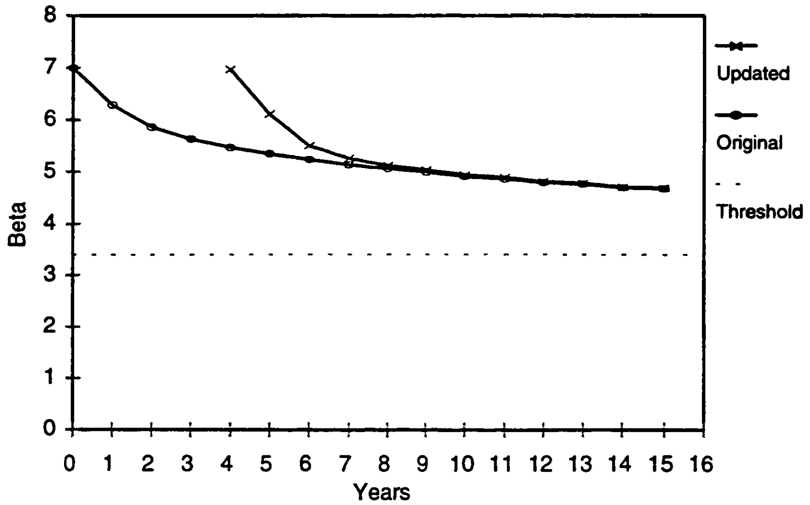
The expected cost of failure tends to have a greater effect on the total costs than the costs of repair and inspection in particular for critical joints. Of the three graphs shown, graph (b) displays the most typical shape based on the results for the many example joints analysed, in that there is a minimum and after this point in time the total costs start increasing.

The expected costs output cannot be compared easily with any other work since few other reliability studies have included expected costs. It is to be noted that the costs vary widely according to the type of joint, whether critical or non-critical, and the probability of failure.

a)



b)



c)

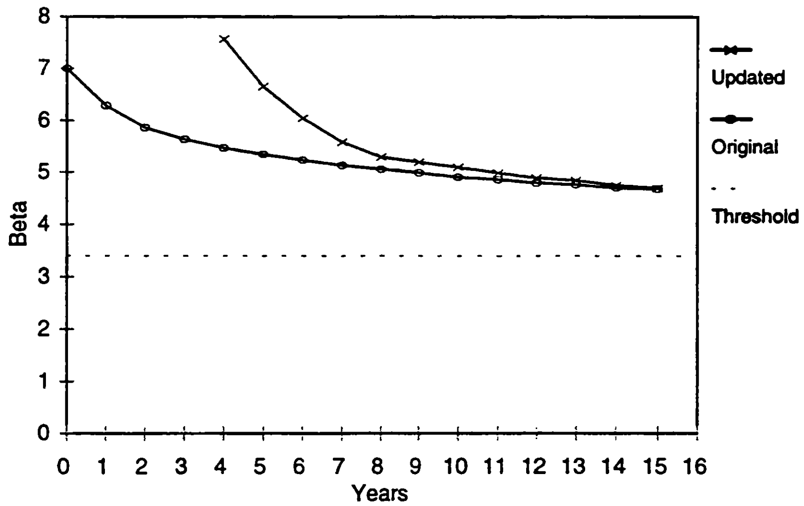


Figure 6.4 Reliability index values

(a) no detection, (b) detection with MPI, (c) detection with ACFM

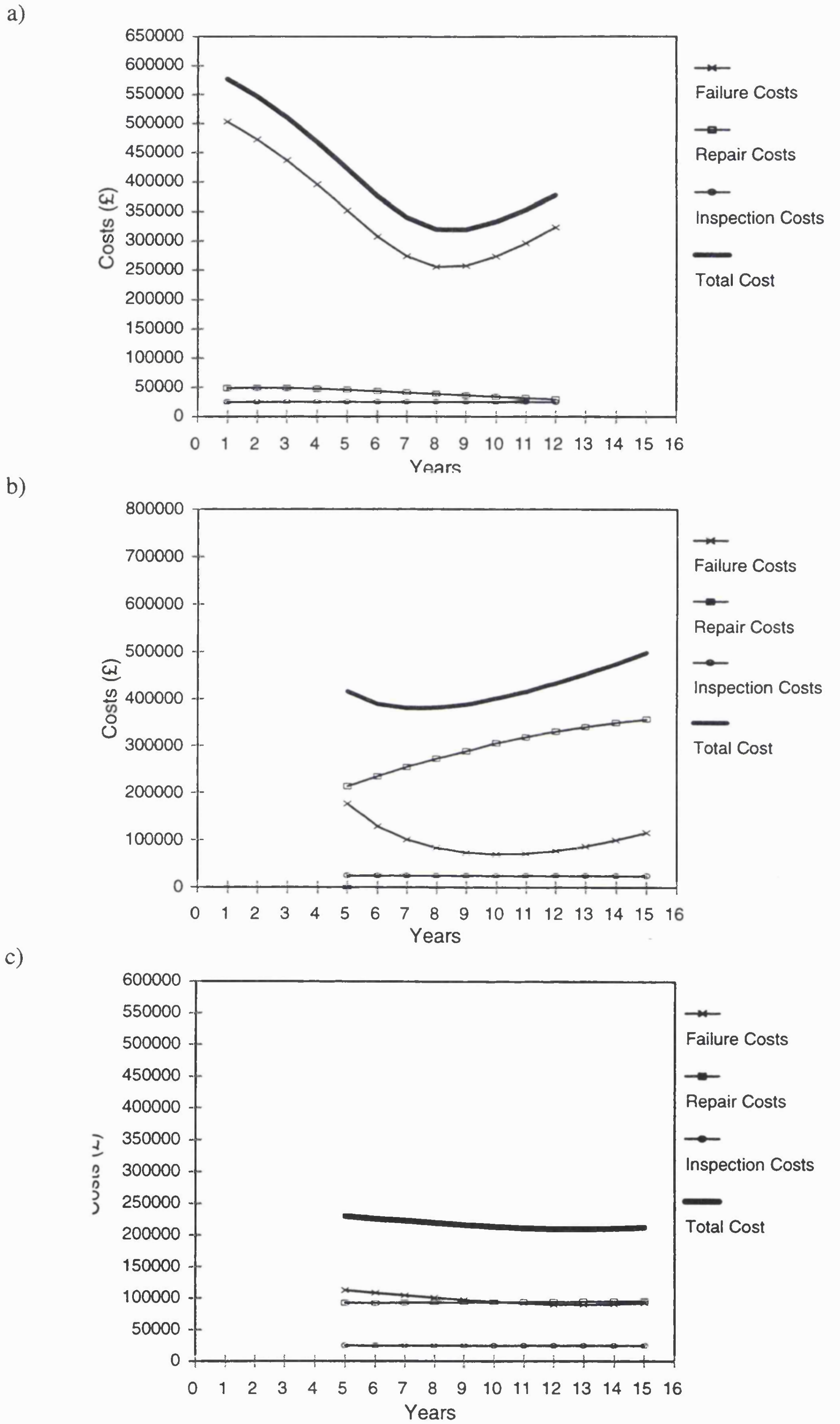


Figure 6.5 Costs for joints with maintenance plans

(a) 01 and CV, weld>0.006, (b) 04 and EC, weld>0.006 (c) 15 and EC, weld>0.002

If only costs are being taken into account, the constraints satisfaction scheduling then need only be considered explicitly for the most expensive joints. The joints for which the expected costs are very low can be included into the final schedule directly, wherever there are resources available. This is because it makes little practical difference to the total costs of the schedule if a joint with maintenance plans costing in the order of 100 times less than the main joints is included in the schedule at its optimum minimal costs period.

6.4 CONSTRAINTS SATISFACTION SCHEDULING IN THE RISC SYSTEM

The use of RISCREL, or any other software providing some form of utility measures, provides the initial data from which scheduling can be carried out taking into account constraints and available resources. This section discusses some of the constraints and then gives possible schedules and a combined schedule based on the expected costs information from RISCREL.

The main constraint is to ensure the integrity of the structure. For this reason, target reliability levels are set. The next constraint is the available time in which to carry out IRM activities. Finally, the estimated total costs are also considered to make a final decision on the best schedule to implement.

6.4.1 Target Safety Levels

In the use of reliability analysis to decide on maintenance actions, it is necessary to set target reliability levels. Although in the RISC methodology decision-making is based mainly on costs evaluation, it is important to be assured that the final schedule of inspections does take threshold minimum reliability levels into account. The possible inspection times based on threshold reliability values are easily decided as was discussed in Section 5.2.4. The difficulty is in deciding what is a reasonable minimum allowable reliability value.

Much work has been carried out in setting appropriate target levels based on diverse considerations such as the consequences of failure, the value of the operation and the public perception of risk. One example of a set of reliability targets is that recommended by the Nordic Committee on Building Regulations (NKB) based on the consequences of failure and the mode of failure (Thoft-Christensen & Baker, 1982). If a component suffers brittle failure resulting in instability, which the NKB termed 'failure type III', then this is more serious than if it suffers ductile failure, and in turn, ductile failure resulting in no reserve strength, termed "type II failure" is more serious than ductile failure with reserve strength due to strain hardening, or type I failure. A component which is important to the structure in that failure may lead to the collapse of the structure requires a higher reliability, relative to a failure for which the consequences are minimal.

The values in the NKB table were chosen to reflect the maximum probability of failure values acceptable. By considering the approximate relationship between reliability index β and the probability of failure POF as explained in Section 3.2 and given as

$$POF \approx 1 - \Phi(\beta) = \Phi(-\beta) \tag{6.3}$$

the corresponding probability of failure values range from 10^{-7} to 10^{-3}.

Table 6.12 Nordic Building Committee reliability index target levels

Failure Consequences	Failure Type I	Failure Type II	Failure Type III
Not serious	3.09	3.71	4.26
Serious	3.71	4.26	4.75
Very serious	4.26	4.75	5.20

In the offshore industry, the target levels for reliability have often been derived by reference to existing structures. An acceptable maximum probability of failure for production structures is considered to be 10^{-4}, which corresponds to a reliability index of 3.72 (MTD, 1989; Diamantidis et al, 1991).

For inspection scheduling based on fatigue fracture as the only failure mechanism, then this corresponds to only one failure type. Thus the set of target reliability values for the RISC methodology requires only three values. If fatigue failure defined as the existence of a through crack is compared to the NBC categories, then as a through-crack almost certainly does not lead to instability and often such a failed joint will still be able to carry some load, fatigue failure may be of type I.

In the RISC project, target reliability indices were proposed by the Registro Italiano Navale (RINA) based on the relative difficulties of accessing the joint for inspection and repair (Dharmavasan et al, 1994b). This criterion is of value at the design stage when deciding the safety factors that should be applied to the design of the component. For instance, if a joint cannot be easily repaired, then it is important to be assured that its reliability is high and hence the design should be appropriate to this requirement. Conversely an easy-to-access joint need not have a large safety factor applied to its strength since it would be possible to monitor it closely and to repair it if and when necessary. The results of inspection and repair can be used to update the reliability indices for the inspected joints and thus ensure the overall reliability of the structure.

There were three categories with appropriate target reliability indices defined and these are given in Table 6.13. RINA proposed that this scheme be applied to inspection planning. Thus, if a joint was considered to be of high importance, then inspection must be carried out before the reliability index falls below a threshold value of 3.4. Similarly, 3.0 and 2.4 represent minimum allowable values for the reliability of moderately important and less important joints.

Table 6.13 Reliability index target levels proposed in RISC

Importance Level	Criteria	Target β
High	members with no access for inspection and repair	3.4
Moderate	members at or below the splash zone	3.0
Low	members above the splash zone	2.4

There is an obvious logical problem with this scheme. The fact that a joint is very difficult to inspect does not increase the need for inspection. Further, if a joint cannot be accessed for inspection, then naturally it cannot be included in a programme of inspections. Even if it assumed that "no access" can be interpreted as "very difficult to inspect", then a definition of "high importance" means only that costs of inspection for these joints are very high.

A scheme for deciding which joints it is important to inspect, irrespective of costs, must instead take into account the relative importance of the joint to the overall reliability of the structure. Redundancy and collapse analyses of a frame structure indicate the level of redundancy and thus identify the critical elements in the structure. Thus it is proposed to define the importance of the joint in terms of its criticality. As the definition of criticality has already been employed to set approximate failure costs for joints, it is proposed by the author that these categories are re-used together with the same target reliability levels as given above as given in Table 6.14 below.

Table 6.14 Reliability index target levels for the case study

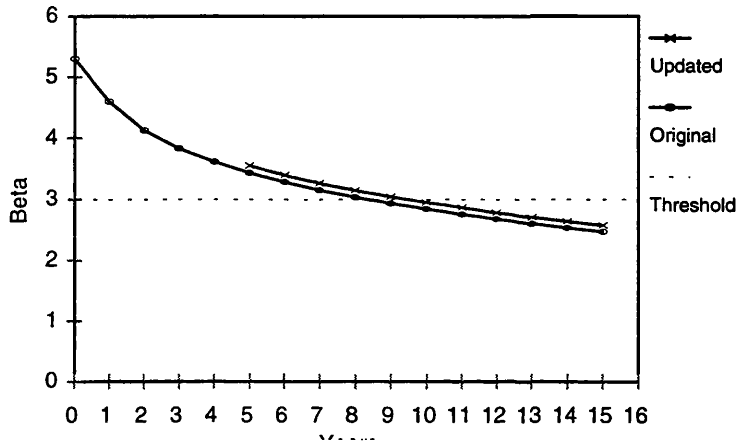
Importance Level	Criteria	Target β
Critical	non-redundant joints	3.4
Non-critical	joints with some redundancy	3.0
Minor	joints which provide some stiffness only	2.4

Although the relative importance of joints is already incorporated into the cost evaluation of inspection plans via the failure costs, these target reliability values ensure that the safety of the structure is not compromised by economic considerations. The target β provides a fail-safe procedure in the selection of the optimal but appropriate inspection plans.

6.4.1.1 Effect on Maintenance Plans

Possible maintenance plans are restricted by the target reliability values. This means that if the β value of the joint falls below the threshold value for that joint just after inspection period IPn, then only maintenance plans for periods up to and before IP_n are allowed.

a)



(b)

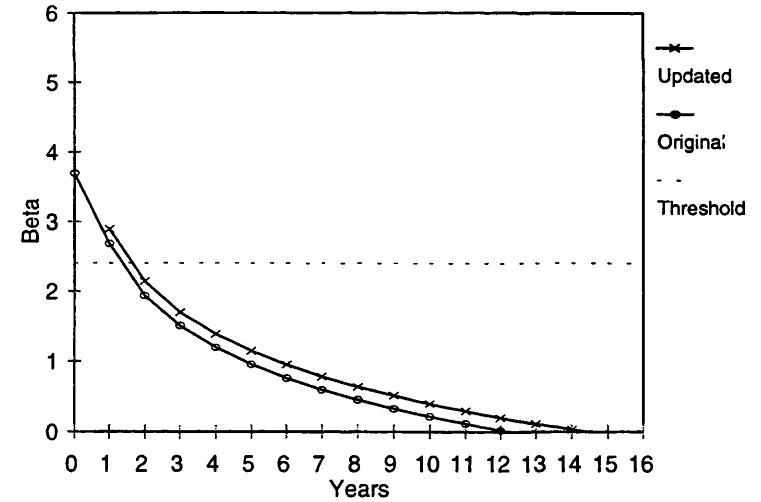


Figure 6.6 Reliability index for joints (a) 13 and (b) 16

An example of this situation arises for joint 13 for which the reliability index, even after updating, falls below the threshold value of 3.0 before year 10. A more extreme example is that of the test joint case 16 of a joint of geometry with a crack detected (and which was not considered for the schedule), which drops to below the threshold value at year 2. This means that for joint 13, inspections should only take place in years 5 to 9 and for joint 16, an inspection must take place immediately.

6.4.2 Using the Scheduling Module

The knowledge base system Scheduling module requires a text file containing the interpreted results from RISCREL in the form of the costs for a subset of possible actions for each joint under consideration. It was found that the expected costs of failure dominated the total costs. Only the joints which had the greatest expected costs were considered. The threshold was set at £100000.

The joints which had to be inspected at certain times, that is those restricted to only one maintenance plan at one inspection period, were ignored in that they would, in practice, be entered into the Scope of Work immediately. The remaining resources are then used to define the search tree for the combined constraints satisfaction scheduling carried out by the RISC KBS. It was found that when considering the threshold set on the maximum allowable expected costs for every maintenance plan, there were no restrictions on the inspection times for some for several joints. These then were also ignored as they can be added to the schedule after the automatic scheduling as required and wherever there are resources available.

The scheduling was based on only one of each joint case requiring inspection, since it is assumed the cracks will not grow in most joints as designed, particularly if they have already been inspected once. All joints which were found to have fatigue cracks in the past, are included in the schedule. In practice, if any unusual findings are made then the schedule would be updated at that point to inspect further sample joints of the same type.

6.4.2.1 Resources for the Search Tree

The search tree is created by allocating objects to the tree nodes (distinct from the nodes of the structure). For the example search, expected costs were the criterion employed for deciding when to inspect. The total resources were taken to be the total time available over the scheduling period. This was set at 15 inspections, thus only fifteen joints were to be considered and these are 01 to 15. In practice, the resource measure would be an average of the number of inspections that can be carried out over five weather windows for a structure; for a structure of 1000+ nodes, corresponding to 3000+ joints, this would be at least in the order of 1000 inspections or 200 inspections per weather window.

The tree was allocated five inspection periods from year 6 to year 10, corresponding to the general UK requirement that a Scope of Work for a five-year period be presented to the certification authorities. For the first combined schedule, each inspection period was allocated resources for three inspections only.

6.4.2.2 Alternative Actions

As explained earlier, joints which must be inspected at a certain time and joints for which many possible alternatives covering all possible inspection periods may be ignored when creating a practical schedule. So, to demonstrate the searching algorithm, from two to four possible actions were considered for each joint. The selected actions are listed below where IP stands for inspection period and the costs have been rounded up to the nearest £1000.

Table 6.15 List of actions for combined scheduling

Joint	Action (Inspection Technique, Repair Criteria)	Cost (£k)	IP
01	MPI, weld when >0.006	230	6
01	ACFM, weld when >0.006	234	7
01	ACFM, weld when >0.006	235	8
02	MPI, weld when >0.005	459	10
02	MPI, weld when >0.005	492	9
02	EC, weld when >0.005	512	7
02	EC, weld when >0.005	520	6
03	MPI, weld when >0.005	388	10
03	MPI, weld when >0.005	401	9
04	ACFM, weld when >0.006	322	6
04	ACFM, weld when >0.006	324	7
04	EC, weld when >0.006	354	8
05	MPI, weld when >0.006	230	6
05	ACFM, weld when >0.006	234	7
05	ACFM, weld when >0.006	235	8
06	MPI, weld when >0.005	478	10
06	MPI, weld when >0.005	501	9
06	EC, weld when >0.005	527	7
07	CV, weld when >0.005	488	6
07	MPI, weld when >0.005	511	10
07	MPI, weld when >0.005	562	9
08	ACFM, weld when >0.006	241	6
08	MPI, weld when >0.006	251	8

Joint	Action (Inspection Technique, Repair Criteria)	Cost (£k)	IP
08	ACFM, weld when >0.006	272	7
09	ACFM, weld when >0.006	435	7
09	MPI, weld when >0.003	479	10
09	MPI, weld when >0.003	498	9
10	EC, weld when >0.005	515	7
10	EC, weld when >0.005	520	6
10	MPI, weld when >0.006	540	6
11	CV, weld when >0.006	324	6
11	CV, weld when >0.006	335	7
11	MPI, weld when >0.005	349	10
12	MPI, weld when >0.001	492	9
12	MPI, weld when >0.001	512	8
12	MPI, weld when >0.001	520	7
13	MPI, weld when >0.005	488	9
13	MPI, weld when >0.005	499	10
13	ACFM, weld when >0.005	520	8
14	ACFM, weld when >0.002	524	7
14	ACFM, weld when >0.002	528	6
14	MPI, weld when >0.002	530	5
15	CV, weld when >0.002	122	6
15	ACFM, weld when >0.006	165	7
15	MPI, weld when >0.005	175	10

6.4.3 Example Programmes of IRM Actions

The initial schedule is made up of the minimal cost actions for each joint and is shown in Table 6.16 overleaf. This schedule has a corresponding estimated total cost of £5729000, given by the sum of the expected cost for each action.

Given that there is a maximum of three inspections allowed for each inspection period, the inspection period 6 has too many inspections allocated to it and inspection periods 8 and 9 are under-utilised. Thus a search is carried out for a schedule which levels out the resource usage.

Table 6.16 Initial schedule of inspections

Year	List of required IRM resources		General region of inspection	
Node-ID	**Joint-ID**	**Inspection Technique**	**Repair Criteria**	**[Cost (£k)]**
Year 6	ACFM, MPI, CV			
1A1	01	MPI	0.006m	230
2A3	04	ACFM	0.006m	322
2A1	05	ACFM	0.005m	230
	07	CV	0.005m	488
	08	ACFM	0.006m	241
	11	CV	0.006m	325
	15	CV	0.002	122
Year 7	ACFM, EC			
	09	ACFM	0.006m	435
	10	EC	0.005m	515
	14	ACFM	0.006m	524
Year 8	**none**			
Year 9	MPI			
	12	MPI	0.005m	492
	13	MPI	0.005m	488
Year 10	MPI, EC			
	02	MPI	0.005m	459
	03	MPI	0.005m	388
	06	EC	0.005m	478

Resources could be re-allocated to the inspection periods which require them. For example in the initial schedule, since no optimal actions exist for inspection period 8, the operator may decide that no inspection should be carried out that year. Instead the resources from year 8 are re-allocated to year 1, and perhaps years 7 and 10. Certainly, resource allocation would need to be carried out in the case that the initial schedule has a positive total resource deficit (TRD), since this indicates that more IRM actions have been proposed than expected in the budget. This is does not occur in this example.

6.4.3.1 Generating Schedules

Three searching techniques have been implemented in the RISC KBS and each one will provide a different proposal for a schedule. The search tree corresponding to the initial schedule is easily built, and is shown below in Figure 6.7. The leaf nodes of the possible actions for only joint 14 are given, as an example, and the year 7 action is in the initial schedule.

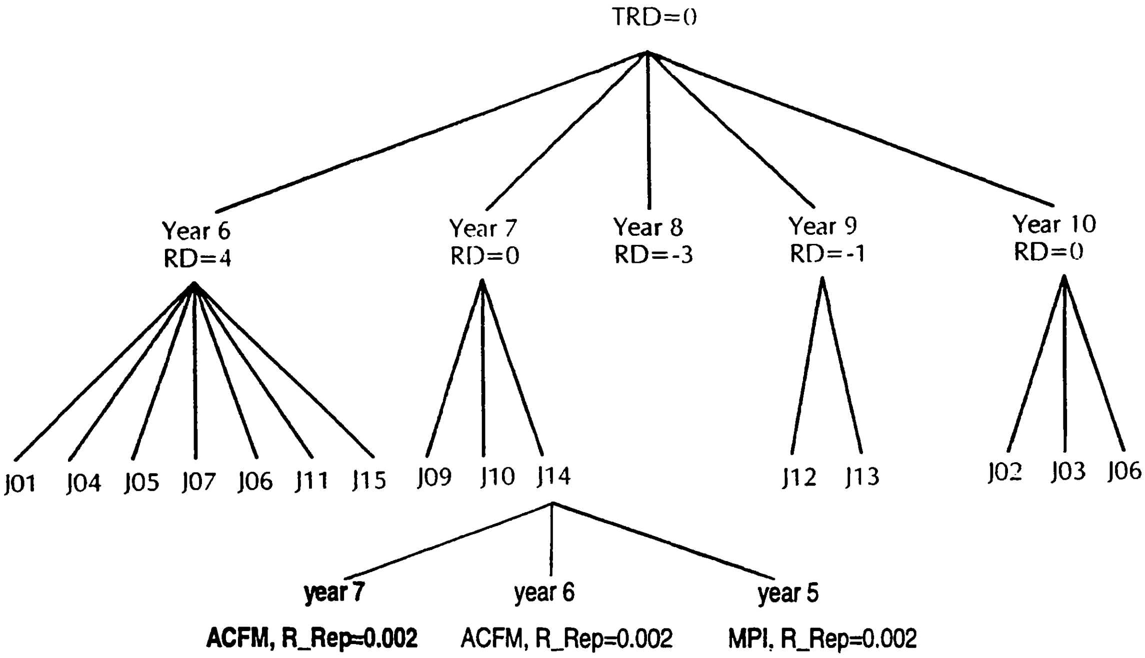


Figure 6.7 Search tree for the initial schedule

Using depth-first searching, the first inspection period, year 6, is considered first as it has a positive resource deficit. Searching this branch, the first joint, 01, is considered for reallocation. Each alternative action for 01 is considered in turn and the action for year 7 is discarded as year 7 has no resources available, so the action for year 8 is selected. Joint 01 is removed from year 6 and reallocated to year 8. As year 6 now has RD=3, the next joint, 04, is considered for reallocation. It can be seen that joint 04 has an alternative action for year 8 and this is selected. The procedure continues in this manner, until year 6 has RD=0.

The search then focuses on the next inspection period with a positive RD, and the search continues down this branch. As it happens, the resources are levelled for this tree once year 6 has been considered. Table 6.17 shows the final allocation of joint inspections to each inspection period and the costs for each action. From this the new total estimated cost is then £5853000.

Using breadth-first search, the target branch changes to the next branch one level up from the point at which the search has failed. The first inspection period to be targeted is also year 6, and the first joint considered is 01. Assuming that the leaf nodes of the tree, that is the actions, are ordered by cost (as in Table 6.16), then the search fails when the first alternative action checked is one for year 7. The next joint, here 04, is considered. The search also fails here as the first alternative is also for a year with non-positive RD; similarly for joints 05 and 07. The first alternative for joint 08 is for year 8 and so joint 08 is re-allocated to year 8. As joints 11 and 15 also have as first alternatives actions for year 7, these are also not re-allocated.

At this point, year 6 has RD = 3 so the search procedure would then focus on the next inspection period with a positive RD. For this example, year 6 is the only year with a positive RD and the search moves back down and considers the second alternatives for each joint. Joints 01 and 04 are re-allocated to year 8, which is then fully utilised, and joint 07 is re-allocated to year 9. The search then ends here as all inspection periods have now non-positive RD values. The proposed schedule is as in the Table 6.18. and the total estimated cost for this proposed schedule is £5862000.

Table 6.17 Schedule of inspections from depth first search

Year	Joints	Cost of IRM Action (k£)
6	08, 11, 15	241+325+122
7	09, 10, 14	435+515+524
8	01, 04, 05	235+354+235
9	12, 13, 07	492+488+562
10	02, 03, 06	459+388+478
Total Cost		5853

Table 6.18 Schedule of inspections from breadth-first search

Year	Joints	Cost of IRM Action (k£)
6	05, 11, 15	234+325+122
7	09, 10, 14	435+515+524
8	08, 01, 04	251+235+354
9	12, 13, 07	492+488+562
10	02, 03, 06	459+388+478
Total Cost		5862

As can be appreciated from the above examples, depth-first and breadth-first methods will give very different results according to the ordering of the nodes in the tree. The assumption has been that the nodes are ordered by name first and then cost second. Variations, such as random ordering perhaps, or reverse ordering, could be tried for alternatives schedules. Best-first searching, however, minimises the effects of the ordering of the nodes.

In best-first searches, the choice of branch depends on the expected result. For this application, there is a look-ahead step which considers which inspection period has greatest resource usage and which alternative action will result in the least increase in total cost for the schedule. The first inspection period to be targeted is the one with the highest RD value, which again is year 6. Of the joints allocated to this inspection period, 04 has an alternative action for year 7 which would increase the total cost by £2000 only. As year 7 has RD = 0, 04 cannot be re-allocated and the next best alternative is considered. There are two possibilities, 01 and 05, each increasing the overall cost by £4000 and both for year and hence cannot be reallocated. Joints 01 and 05 also have alternatives with increased costs of £5000, both for year 8, so both are reallocated as year 8 has RD<-2. The next re-allocations are of joint 08 to year 8 with an increased cost of £10000 and joint 07 to year 9 with increased cost of £74000. This proposed schedule is shown in Table 6.19 and has total estimated costs of £5823000.

Table 6.19 Schedule of inspections from best-first search

Year	Joints	Cost of IRM Action (k£)
6	04, 11, 15	322+325+122
7	09, 10, 14	435+515+524
8	01, 05, 08	235+235+251
9	12, 13, 07	492+488+562
10	02, 03, 06	459+388+478

This procedure may be modified for the short, but wide search trees that are constructed for this application to allow the best of all the alternatives, irrespective of the inspection period to be considered for re-allocation. For this specific example, however, such a procedure would make little practical difference.

6.4.3.2 Heuristics-based Searching and Multiple Constraints

The scheduling above has only taken into account costs and time constraints. No other constraint other than a maximum number of inspections that can be carried out in one weather window was considered in the simple example above. Examples of other constraints to be considered include

1. Costs, as well as time, may be restricted in any one weather window.

2. There may be restrictions on the use of NDT or repair equipment, so that if an inspection task requires the use of ACFM equipment, say, then it should not be the only such task and furthermore, a good proportion, if not all of the other ACFM inspection tasks should also be carried out during the same weather window.

3. Joints on one node and neighbouring nodes need to be inspected at the same time to avoid moving unnecessarily the inspection vessel in any one weather window.

To allow consideration of general soft and quantifiable resources, a metric for the vector of resource deficits for each inspection period i $\underline{RD}(i)$ can be defined. This is a weighted linear sum of the resources which will favour preferred constraints and which can then replace the cost value as the criterion for the best-first search.

For non-quantitative constraints, such as "ACFM is the preferred inspection technique", a heuristic search strategy was devised to find a best solution in terms of cost or other criteria defined using heuristics about constitutes a good schedule. For neighbouring joints to be taken into account, some way of identifying geographical proximity is required. This may be carried out by the use of heuristics based on the structural model or by the use of a metric representing the distance of the joint from other joints.

Figure 6.8 shows the heuristic-based search method written in ProTalk incorporating a total resource cost check into a breadth-first search method. The function *focus_on_next _action*, with functions *focus_on_next_action_period* and *focus_on_next_component* embedded within it, forces the search from upper to lower branches of the tree. Function *update_schedule* is the constraint propagation procedure used to update the schedule and to recalculate resource deficits.

At the component level, before the system searches breadth-first through the tree, all the component nodes are sorted by the function *SortBySlot,* to ensure that the less expensive actions are considered first. Thus this search procedure is in effect a variation on the best-first search. An extension to the RISC System would be to allow *SortBySlot* to order components according to some other criteria should another analysis program, other than RISCREL, providing another measure of utility for an action be incorporated into the system.

The important function to note is *test_candidate_action* as this invokes a set of rules to check the constraints for the possibility of using the tested action. In this way, the best-first search algorithm has been generalised to allow a combination of quantitative criteria and heuristics to be used to define the "best" node to consider next.

```
function heuristic_based_search_method ()
{for ?action_period inlist all instanceof action_periods;
   do { ?k = 1;
   while {?k< init_schedule.no_of_total_alternative_actions;
                                        ?action_period.resource_deficit > 0;}
      do { ?candidate_action_list = Null;
         for ?component inlist all ?action_period.components;
            do {focus_on_next_action(?component); ?k = ?k + 1;}
         for ?component inlist all ?action_period.components;
            do{?n1 = ?component.current_action_no;
               ?n2 = ?component.previous_action_no;
               ?current_action = ListNth(all ?component.actions, ?n1);
               ?previous_action = ListNth(all ?component.actions, ?n2);
               ?component.cost_increase = ?current_action.cost -
                                                previous_action.cost;};
         SortBySlot(all ?action_period.components,cost_increase,>");
         for ?components inlist all ?action_period.components;
            do{if ?component.current_action_no <
                                      ?component.total_action_no;
               then{?action=ListNth( all ?component.actions,
                                      ?component.current_action_no);
               if test_candidate_action(?action_period, ?component,
                                                ?action)== TRUE;
               then update_schedule(?action_period, ?component, ?action);}
}}}
```

Figure 6.8 Heuristics-based search algorithm

6.4.3.3 Selection of a Programme

The best-first algorithm can reduce the cost of a new schedule significantly when the numbers of actions and components are large, compared with normal depth-first or breadth-first search methods. Nevertheless, it is important to remember that the costs may vary according to the number joints which can be inspected in one shift by one diver, and hence the total costs, whether the output of cost evaluation or based on a combined total resource check, are only an estimate of the cost. The use of several searching techniques provides several usable schedules.

These simple searching techniques provide a powerful way of providing the expert user with several possible rational alternative schedules from the RISCREL cost evaluation output. The final decision would depend on the operator's in-depth knowledge of the resources which may be varied based on the initial schedules found by the RISC KBS. The changes in the resources can then fed back into the RISC KBS to allow the operator to view how these changes affect the expected costs for a schedule, as part of a "what-if" analysis.

6.5 DISCUSSION

The case presented here shows how RISCREL results may be used to provide rational schedules based on costs and time constraints. The procedure is a generalised scheduling algorithm which may be extended to allow any combination of resources and soft resource constraints to be considered.

Hard constraints are assumed to have been dealt with before constructing the initial scheduling tree. For instance, a hard constraint which has already been considered is that of safety: inspections must take place before the reliability index falls below some threshold value. Other hard constraints, such as "all damaged joints must be inspected immediately", are dealt with very simply: the actions are added to the tree and no alternatives are allowed.

In general, incorporating any hard constraints requires modifications to the tree by eliminating non-permissable actions and adding must-happen actions. At the moment, hard constraints are dealt with by the ad hoc application of heuristics. No facility has yet been provided for updating the scheduling search tree in a systematic way. Furthermore, the scheme does not include ways to handle overlong searches or the situation that there are no solutions. To overcome both of these problems automatically may require more advanced features such as constraint network analysis combined with constraint relaxation techniques. It may be that only a simple check with the user for permission to continue or quit the searching is required. Another simple facility is to allow the user to make changes to the resource allocation dynamically, that is, part way through a search.

The case study highlights the data required for the reliability analysis and some of the issues. This example of the searching algorithms in use illustrates their effectiveness. The next step to develop this scheme further would be to trial the RISC System within an operator organisation and observe maintenance engineers using the Constraints Satisfaction (CS) Scheduling module. From such an exercise, it will be possible to formulate the requirements for more advanced automatic intelligent features to support their decisions.

Table 6.20 RISCREL input data

Structure	Variable	Type	Description	RISCREL	X	Dist	M/P	Remarks and KBS Derivation of mean value	2nd M/P	3rd M/P	4th M/P
Joint_Geometry	Member_Thickness	double	Chord or brace thickness in meter units	T_CHORD	1	3	m	if Crack.Location = Brace, then T_CHORD = Brace.Thickness, else T_CHORD = Chord.Thickness	d10%	0	0
Crack_Geometry	Initial_Depth	double	Initial crack depth or at last inspection	A0	2	3	m	TNO to recommend initial value. if Joint.Inspected = Yes then A0 = Joint.LastInspection.Crack.Depth else A0 = Structure.InitialCrackDepth (default 0.15 mm)	d10%	0	0
Crack_Geometry	Initial_Length	double	Initial crack half length or at last inspection	C0	3	3	m	TNO to recommend initial values. if Joint.Inspected = Yes then C0 = Joint.LastInspection.Crack.Length else A0 = Structure.InitialCrackHLength (default 0.75mm)	d10%	0	0
Crack_Geometry	Member_Thickness_AWR	double	Member thickness after weld repair	T_R1	4	3	m	Same value as member thickness. = T_CHORD	d10%	0	0
Crack_Geometry	Initial _Depth_AWR	double	Initial crack depth after weld repair	A0_R1	5	3	m	TNO to recommend. default 0.15 mm	d10%	0	0
Crack_Geometry	Initial_Length_AWR	double	Initial crack length after weld repair	C0_R1	6	3	m	TNO to recommend. default 0.75mm	d10%	0	0
Crack_Geometry	Member_Thickness_AGR	double	Member thickness after Grind repair	T_R2	7	3	m	RINA to recommend. = (T_CHORD - Grind.Depth) and Grind.Depth default 2 mm	d10%	0	0
Crack_Geometry	Initial _Depth_AGR	double	Initial crack depth after Grind repair	A0_R2	8	3	m	TNO to recommend. See reference in IMREL manual. default 0.10 mm	d10%	0	0
Crack_Geometry	Initial_Length_AGR	double	Initial crack length after Grind repair	C0_R2	9	3	m	TNO to recommend. Reference in IMREL manual. default 0.10 mm	d10%	0	0
Load_Input	WASR	double	Weighted Average Stress Range (MPa)	WASR	10	2	m	UCL to carryout study. Set-Up values.	d10%	0	0

Structure	Variable	Type	Description	RISCREL	X	Dist	M/P	Remarks and KBS Derivation of mean value	2nd M/P	3rd M/P	4th M/P
Load_Input	Weibull_A	double	Weibull-A Parameter to calculate Exceedence Curve	Weibull_A	11	3	m	UCL to carryout study. Set-Up values.	d10%	0	0
Load_Input	Weibull_B	double	Weibull-B Parameter to calculate Exceedence Curve	Weibull_B	12	3	m	UCL to carryout study. Set-Up values.	d10%	0	0
Material_Input	Paris_C	double	Paris Constant C 1 for the material	Paris_C1	13	3	m	From HLMDB	DB	DB	DB
Material_Input	Paris_C	double	Paris Constant C 2 for the material	Paris_C2	14	3	m	From HLMDB	DB	DB	DB
Material_Input	Paris_C	double	Paris Constant C 3 for the material	Paris_C3	15	3	m	From HLMDB	DB	DB	DB
Material_Input	Paris_C	double	Paris Constant C 4 for the material	Paris_C4	16	3	m	From HLMDB	DB	DB	DB
Material_Input	Paris_C	double	Paris Constant C 5 for the material	Paris_C5	17	3	m	From HLMDB	DB	DB	DB
Material_Input	Paris_C_AWR	double	Paris Constant C 1 after weld repair	Paris1	18	3	m	= Paris_C1	DB	DB	DB
Material_Input	Paris_C_AWR	double	Paris Constant C 2 after weld repair	Paris2	19	3	m	= Paris_C2	DB	DB	DB
Material_Input	Paris_C_AWR	double	Paris Constant C 3 after weld repair	Paris3	20	3	m	= Paris_C3	DB	DB	DB
Material_Input	Paris_C_AWR	double	Paris Constant C 4 after weld repair	Paris4	21	3	m	= Paris_C4	DB	DB	DB
Material_Input	Paris_C_AWR	double	Paris Constant C 5 after weld repair	Paris5	22	3	m	= Paris_C5	DB	DB	DB
Weld_Geometry	Weld_Toe_Radius	double	Weld toe radius in meter units	W_T_R	23	2	m	Set-Up value. default 0.5 mm	d10%	0	0
Weld_Geometry	Weld_Length	double	Weld leg length in m units	W_LEN	24	2	m	RINA Function of member thickness and weld angle. default '= SIN(W_ANG) x Brace.Thickness	d10%	0	0

Structure	Variable	Type	Description	RISCREL	X	Dist	M/P	Remarks and KBS Derivation of mean value	2nd M/P	3rd M/P	4th M/P
Weld_Geometry	Weld_Angle	double	Weld angle in degrees	W_ANG	25	2	m	RINA Range of values 45 to 70 deg =Joint.WeldAngle	d10%	0	0
Weld_Geometry	Weld_Toe_Radius_AGR	double	Weld toe radius in meter units for Grind repair	W_T_R_G	26	2	m	RINA 4 to 5 mm. default = 4mm or else = GrindInstrument.Radius	d10%	0	0
Weld_Geometry	Weld_Length_AGR	double	Weld leg length in m units for grind repair	W_LEN_G	27	2	m	= W_LEN	d10%	0	0
Weld_Geometry	Weld_Angle_AGR	double	Weld angle in degrees for grind repair	W_ANG_G	28	2	m	= W_ANG	d10%	0	0
Weld_Geometry	Weld_Toe_Radius_AWR	double	Weld toe radius in meter units for weld repair	W_T_R_W	29	2	m	= W_T_R	d10%	0	0
Weld_Geometry	Weld_Length_AWR	double	Weld leg length in m units for weld repair	W_LEN_W	30	2	m	RINA Longer than W_LEN . = W_LEN + WeldRepair.AddedLength	d10%	0	0
Weld_Geometry	Weld_Angle_AWR	double	Weld angle in degrees for weld repair	W_ANG_W	31	2	m	= W_ANG	d10%	0	0
Inspection_Method	Perf_Epsilon1	double	Performed Inspection Uncertainty for method 1	Epsilon1	32	2	m	WDD to recommend MPI value. From inspection DB	DB	DB	DB
		double		Epsilon2	33			NOT Used			
		double		Epsilon3	34			NOT Used			
		double		Epsilon4	35			NOT Used			
		double		Epsilon5	36			NOT Used			
		double		Epsilon6	37			NOT Used			
Inspection_Method	Perf_POD1	double	Performed Inspection POD for method 1	POD1	38	4 or 9	p	WDD to recommend MPI value. From inspection DB	DB	DB	DB
		double		POD2	39			NOT Used			
		double		POD3	40			NOT Used			
		double		POD4	41			NOT Used			

Structure	Variable	Type	Description	RISCREL	X	Dist	M/P	Remarks and KBS Derivation of mean value	2nd M/P	3rd M/P	4th M/P
		double		POD5	42			NOT Used			
		double		POD6	43			NOT Used			
Inspection_Method	Plan_Epsilon1	double	Planned Inspection Uncertainty for method 1	Epsilon_P1	44	2	m	WDD to recommend MPI value. From inspection DB	DB	DB	DB
		double		Epsilon_P2	45			NOT Used			
		double		Epsilon_P3	46			NOT Used			
		double		Epsilon_P4	47			NOT Used			
		double		Epsilon_P5	48			NOT Used			
		double		Epsilon_P5	49			NOT Used			
Inspection_Method	Plan_POD1	double	Planned Inspection POD for method 1	POD_P1	50	4 or 9	p	WDD to recommend MPI value. From inspection DB	DB	DB	DB
	Plan_POD2	double		POD_P2	51			NOT Used			
	Plan_POD3	double		POD_P3	52			NOT Used			
	Plan_POD4	double		POD_P4	53			NOT Used			
	Plan_POD5	double		POD_P5	54			NOT Used			
	Plan_POD6	double		POD_P6	55			NOT Used			
R6_Data	Fracture_Toughness	double	Fracture toughness of the material	SKIC	56			NOT Used			
R6_Data	Yield Stress	double	Yield stress	SIGY	57			NOT Used			
R6_Data	Residual_Stress	double	Residual Stress	SIGR	58			NOT Used			
R6_Data	HS	double	Significant Wave height	HS	59			NOT Used			
R6_Data	TZ	double	Zero Crossing Period	TZ	60			NOT Used			
R6_Data	SS	double	Stress Process	SS	61			NOT Used			

The following lists the parameters used in RISCREL.

Structure	Variable	Type	Description	RISCREL	PVEC	Remarks	KBS Derivation of mean value
Analysis_Control	Analysis_Status	enum	1= Initialise, 2= Normal, 3= Error, 4= Internal Use	AS	1	Used internally.	1
Analysis_Control	Error_Returned	integer	Error Code	EC	2	Used internally	0
Joint_Geometry	Critical_C_Depth_Ratio	double	Critical Crack depth as a percentage of member thickness	CR_C_D	3		default 90
Load_Input	Loading_Method	enum	1= WASR, 2= Weibull, 3=SRPD	L_METH	4	Option 3 not implemented also current usage has different options number	if Joint.LoadData.Method = WASR then L_METH = 1 else if Joint.LoadData.Method = Weibull then = 2 else =3
Load_Input	Cycles_Year	double	Number of Cycles per year	CYC_Y	5	Obtained from ULDAN	value set at Set-Up
Load_Input	Bias_Factor	double	Weibull Bias Factor	BIAS_F	6	Setup Load.WeibullBiasFactor	default 1
CFA_Input	Depth_Growth_Law	enum	1 = Linear Paris , 2= Threshold Paris, 3= Forman Paris	D_G_L	7	Currently only options 1 and 8 are being used	default 1
CFA_Input	Length_Growth_Law	enum	1= Linear Paris, 2=Threshold Modification, 3=Forman Modification, 4=Forcing Function, 5=Tubular X-joint Forcing Function, 6=Vosikovsky, 7=User defined, 8 = new option	L_G_L	8	Currently only option 1 is used	default 1 but if SIF_M = 1 or 2 then L_G_L = 0
CFA_Input	No_of_Segments	integer	Number of segments (1 to 5)	D_SEG	9		From HLMDB
CFA_Input	Paris_m	double()	Paris exponent m1 for the material	PAR_M1	10		From HLMDB
CFA_Input	Paris_m	double()	Paris exponent m2 for the material	PAR_M2	11		From HLMDB
CFA_Input	Paris_m	double()	Paris exponent m3 for the material	PAR_M3	12		From HLMDB
CFA_Input	Paris_m	double()	Paris exponent m4 for the material	PAR_M4	13		From HLMDB
CFA_Input	Paris_m	double()	Paris exponent m 5 for the material	PAR_M5	14		From HLMDB
CFA_Input	C1_Factor	double	C1 Factor for Quadratic Forcing Function (Dijkstra)	C1	15	Required for option 4 of L_G_L	default 0

Structure	Variable	Type	Description	RISCREL	PVEC	Remarks	KBS Derivation of mean value
CFA_Input	C2_Factor	double	C2 Factor for Quadratic Forcing Function (Dijkstra)	C2	16	Required for option 4 of L_G_L	default 0
CFA_Input	X_Joint_Diameter	double	Diameter of Tubular X-joint Forcing Function (TNO)	X_J_DI	17	Required for options 5 or 8 of L_G_L	default 0
CFA_Input	ar	double()	User defined forcing function array	Ar1	18	Required for option 7 of L_G_L	default 0
CFA_Input	ar	double()	User defined forcing function array	Ar2	19	Required for option 7 of L_G_L	default 0
CFA_Input	ar	double()	User defined forcing function array	Ar3	20	Required for option 7 of L_G_L	default 0
CFA_Input	ar	double()	User defined forcing function array	Ar4	21	Required for option 7 of L_G_L	default 0
CFA_Input	ar	double()	User defined forcing function array	Ar5	22	Required for option 7 of L_G_L	default 0
CFA_Input	cr	double()	User defined forcing function array	Cr1	23	Required for option 7 of L_G_L	default 0
CFA_Input	cr	double()	User defined forcing function array	Cr2	24	Required for option 7 of L_G_L	default 0
CFA_Input	cr	double()	User defined forcing function array	Cr3	25	Required for option 7 of L_G_L	default 0
CFA_Input	cr	double()	User defined forcing function array	Cr4	26	Required for option 7 of L_G_L	default 0
CFA_Input	cr	double()	User defined forcing function array	Cr5	27	Required for option 7 of L_G_L	default 0
CFA_Input	SIF_Method	enum	1= AVS, 2= TPM, 3=Holdbrook-Dover, 4=Newman-Raju, 5=Srawley-Brown-Orange-Wilson	SIF_M	28	If 1 or 2 then L_G_L is not necessary	default 4
CFA_Input	BM_Relationship	enum	1=No decay, 2= Linear bending decay, 3=Quadratic Bending decay	B_M_Eq	29		default 1

Structure	Variable	Type	Description	RISCREL	PVEC	Remarks	KBS Derivation of mean value
CFA_Input	Initial_BM_Ratio	double	Initial Bending to Membrane stress ratio	I_BM_R	30		default 1.0
CFA_Input	Pure_Bending	double	Pure bending case	P_BEND	31	If this is 1 then I_BM_R not used	default 0
CFA_Input	SCF_Hot_Spot	double	Hotspot Stress Concentration Factor	HS_SCF	32	Only used by AVS/TPM	if SIF_METHOD = 1 or 2 then = Joint.LoadData.HotSpotSCF
CFA_Input	SCF_Average	double	Average Stress Concentration Factor	AV_SCF	33	Only used by AVS/TPM	if SIF_METHOD = 1 or 2 then = Joint.LoadData.AverageSCF
CFA_Input	Geometry_Parameter	double	Beta ratio of the Joint for TPM or AVS method	GEO_P	34	Only used by AVS/TPM	if SIF_METHOD = 1 or 2 then = Joint.Beta (Joint.Beta = Joint.BraceThickness/ Joint.ChordThickness)
CFA_Input	MK_Factor	enum	1= Plate, 2=IIW-PD6493, 3=DIANA-T/OMAE, 4=Modified IIW-PD6493, 5=DIANA-T/WIDTH OMAE93	MK_FAC	35	Currently only options 3 or 4 are used	= Joint.CFAnalysis.MK_Factor
CFA_Input	Half_Plate_Width	double	Half Plate Width for SIF calculation of Finite Width plate	H_P_WI	36	TNO Usually infinity (which is input as 0)	default 0
CFA_Input	3D_MRF_OMMA	double	3D Membrane reduction for depth Mk-OMMA	OMMA	37		default 0.9
CFA_Input	3D_MRF_OMMC	double	3D Membrane reduction for length Mk-OMMC	OMMC	38		default 0.8
CFA_Input	3D_MRF_OMBA	double	3D Bending reduction for depth Mk-OMBA	OMBA	39		default 0.9
CFA_Input	3D_MRF_OMBC	double	3D Bending reduction for Length Mk-OMBC	OMBC	40		default 0.8
CFA_Input	Integration_Control	enum	1=Percentage increment, 2=Tolerance	INT_C	41	TNO Always , but error in code whereby 0 = %, 1 = tolerance	default 0 (see error)
CFA_Input	Increment_Percent	double	Increment percentage for integration	ALPHA	42	Not used if INT_C = 2	default 5
CFA_Input	Tolerance	double	Tolerance	TOL	43	Not used if INT_C = 1	default 0

Structure	Variable	Type	Description	RISCREL	PVEC	Remarks	KBS Derivation of mean value
CFA_Input	Threshold_Mod_Depth	double	Threshold value for Depth direction	TH_M_D	44	TNO Not Used	default 0
CFA_Input	Forman_Depth_1	double	Forman Modification Factor Kcr for Depth	FOR_D1	45	TNO Not Used	default 0
CFA_Input	Forman_Depth_2	double	Forman Modification Factor R for Depth	FOR_D2	46	TNO Not Used	default 0
Plan_Input	Design_Year	integer	Design Life of the structure in years	L_TIME	47	Given with structure	default 25
Plan_Input	Depth_Last_Inspection	double	Depth at Last inspection	A_INSP	48	Stored with inspection history of joint	= A0
Plan_Input	Year_of_Inspection	double	Year of Inspection	Y_INSP	49	Stored with inspection history of joint	
Plan_Input	Year_Plan_Inspection	double	Year of Planned Inspection	Y_PLAN	50	Sensitivity study carried out on this parameter	default 5
Plan_Input	Repair_Criteria	integer	Crack depth of Repair criteria	A_REP	51	From list of maintenance plans generated for joint	=GetArg1FromArg2(RepairCriteria , NextOneFrom(Joint.ListOfMPs))
Plan_Input	Cost_of_Failure	double	Cost of failure	C_FAIL	52	Given for each node	= Node.CostOfFailure
Plan_Input	Cost_of_Grinding	double	Cost of Repair 1, Grinding	C_REP1	53	Given for each node	= Node.CostGrindRepair
Plan_Input	Cost_of_Welding	double	Cost of Repair 2, Welding	C_REP2	54	Given for each node	= Node.CostWeldRepair
Plan_Input	Cost_of_Inspection1	double	Cost of Inspection of Inspection type 1	C1_INS	55	From inpection DB, and from list of maintenance plans generated for joint	=GetArg1FromArg2(InspectionTecnique , NextOneFrom(Joint.ListOfMPs)) . Cost
Plan_Input	Cost_of_Inspection2	double	Cost of Inspection of Inspection type 2	C2_INS	56	Not used	0
Plan_Input	Cost_of_Inspection3	double	Cost of Inspection of Inspection type 3	C3_INS	57	Not used	0
Plan_Input	Cost_of_Inspection4	double	Cost of Inspection of Inspection type 4	C4_INS	58	Not used	0
Plan_Input	Cost_of_Inspection5	double	Cost of Inspection of Inspection type 5	C5_INS	59	Not used	0

Structure	Variable	Type	Description	RISCREL	PVEC	Remarks	KBS Derivation of mean value
Plan_Input	Cost_of_Inspection6	double	Cost of Inspection of Inspection type 6	C6_INS	60	Not used	0
Plan_Input	Per_Insp_Type	integer	Performed Inspection Type(Method 1 to 6)	I_TYPD	61	Set to 1	1
Plan_Input	Plan_Insp_Type	integer	Planned Inspection Type (Method 1 to 6)	I_TYPP	62	Set to 1	1
Plan_Input	Present_Capital	double	Present Year of Capitalisation	Y_CAPI	63	Given with structure	=Structure.YearOfCapitalisation
Plan_Input	Rate_of_Interest	double	Real Rate of Interest	R_RATE	64	Given with structure	=Structure.InflationRate
R6_Data	Wave_Direction_R6	double	Wave direction for R6 Failure	W_DIR	65	Not used	0
R6_Data	Zero_Cross_Period	double	Zero crossing Period in seconds	TZ	66	Not used	0

The following are RISCREL analysis control parameters, not used by the limit state function.

Structure	Variable	Type	Description	RISCREL	KBS Derivation of Value
Analysis_Control	IMTask	integer	RISCREL control parameter Task: 0=Updated Reliability, 1= Cost Evaluation	IMTask	For Scheduling =1, At Setup and at updating from Observations = 0
Analysis_Control	INSP	integer	RISCREL control parameter Inspection event: 1 =(crack size= observed size), 2=(crack<observed), 3=(crack>observed), 4 = no detection	INSP	if no detection or no inspection then INSP=4 else default INSP =1

Codes used in the above tables are:

Column Name	Code	Meaning
Dist	2	Normal
Dist	3	Lognormal
Dist	4	Exponential
Dist	9	Weibull
M/P	m	Moments are used for the distribution description.
M/P	p	Parameters are used for the distribution description.
2nd M/P	d00%	Default value is 00% of mean value.

7 AUTOMATED INSPECTION

This chapter addresses the use of intelligent systems for the automatic interpretation of inspection results. Since data on the reliability of an inspection method and the interpretations of inspection results are important as input to the reliability analysis, it is valuable to consider ways in which the reliability of any inspection technique may be improved and the interpretation of an inspection result made be made more consistent.

The requirements of the project AIRES (Automated Image Recognition using Expert Systems) are given. The aim of the AIRES project was to provide the technology for automated total surface inspection of components or materials during both production and service by using an expert system framework to provide image reconstruction from multiple sensors. For this an integrated software and multi-sensor system was developed and this is next described. Details of the knowledge representation for the AIRES system are given. Finally, the shortcomings of the approach to the AIRES work together with aspects of data fusion and knowledge representation are outlined.

7.1 INTRODUCTION TO THE AIRES SYSTEM

The primary requirements of an automated inspection system are to identify defects, surface condition, roughness and material quality. For this, several different non-destructive testing (NDT) methods are best used to characterise the surface defects, such as cracks or pits, or the general quality of component surfaces, such as corrosion.

To automate the inspection process requires an understanding the behaviour of the sensors and methods of interpreting the sensor data. It was found in the work here described that it is not necessarily a trivial exercise to interpret the results obtained from an NDT tool. Formal interpretation procedures may exist, but may be based on idealised conditions and in practice human experts will modify the procedures according to the situation. In addition, when trying to make definite decisions about the existence of flaws, it is necessary to use more than one NDT method, often based on two different physical principles, such as magnetic particle inspection and ultrasonics, or close visual inspection and the ACFM technique.

The Automated Image Reconstruction using Expert Systems, AIRES, system was developed to combine vision and electromagnetic sensors to locate, classify and characterise surface flaws in automobile components. The interpretation of the information gathered from the sensors was carried out by specially developed software modules which were integrated into an expert system development tool based on the blackboard model. The AIRES software is a blackboard-based expert system which carries out interpretation of the signals from a visual (VS) sensor and an electromagnetic (EM) sensor. The electromagnetic sensor is an alternating current field measurement (ACFM) probe (Collins et al, 1990).

The following sections describe the software architecture used to integrate the different areas of expertise necessary to provide automated defect classification and characterisation using electromagnetic sensors.

7.1.1 The Electromagnetic ACFM Inspection Technique

The electromagnetic sensor used in this work was based on the alternating current field measurement (ACFM) technique (Dover & Collins, 1980; Collins et al, 1990). This technique was developed at University College London for the detection and sizing of surface cracks and defects in metallic structures and components.

Under appropriate conditions, if an alternating current is induced in a surface, the magnetic field above the surface will vary according to the geometry of the surface. To investigate the geometry of a component or structure, and thus to detect and size cracks and other surface defects, a current is induced in the surface, usually by an inducing coil suspended above it, and the magnetic field (**B**) over the surface is then recorded.

7.1.1.1 The Electromagnetic Field Data

There is a broad correlation between the nature of the magnetic field and the features of the surface. An example of the magnetic field components, $\mathbf{B}_x$, $\mathbf{B}_y$, $\mathbf{B}_z$, in the three axis x, y and z respectively as predicted by electromagnetic theory for a crack of length 8mm and depth 1mm in a flat surface is shown in Figure 7.1, where the unit of the x and y axes are 1mm. Typically, human experts are able to identify defects from such graphs of sensor data. It is the task of the AIRES software to automate the use of **B** data to determine the surface characteristics.

7.1.1.2 The ACFM Sensor

The sensor used to record the magnetic field data is illustrated in Figure 7.2. An inducing solenoid is suspended above an array of magnetic field sensors, which are cylindrical coils arranged in a 10 by 12 array. A coil can only record the field in the direction of its axis, so they are placed as shown: alternating in each of the three x, y and z directions. To record a complete set of 10 by 10 magnetic field data, the sensor is placed in three adjacent positions on the surface. The electromagnetic sensor outputs matrices giving the x, y, and z components of the magnetic field over the surface.

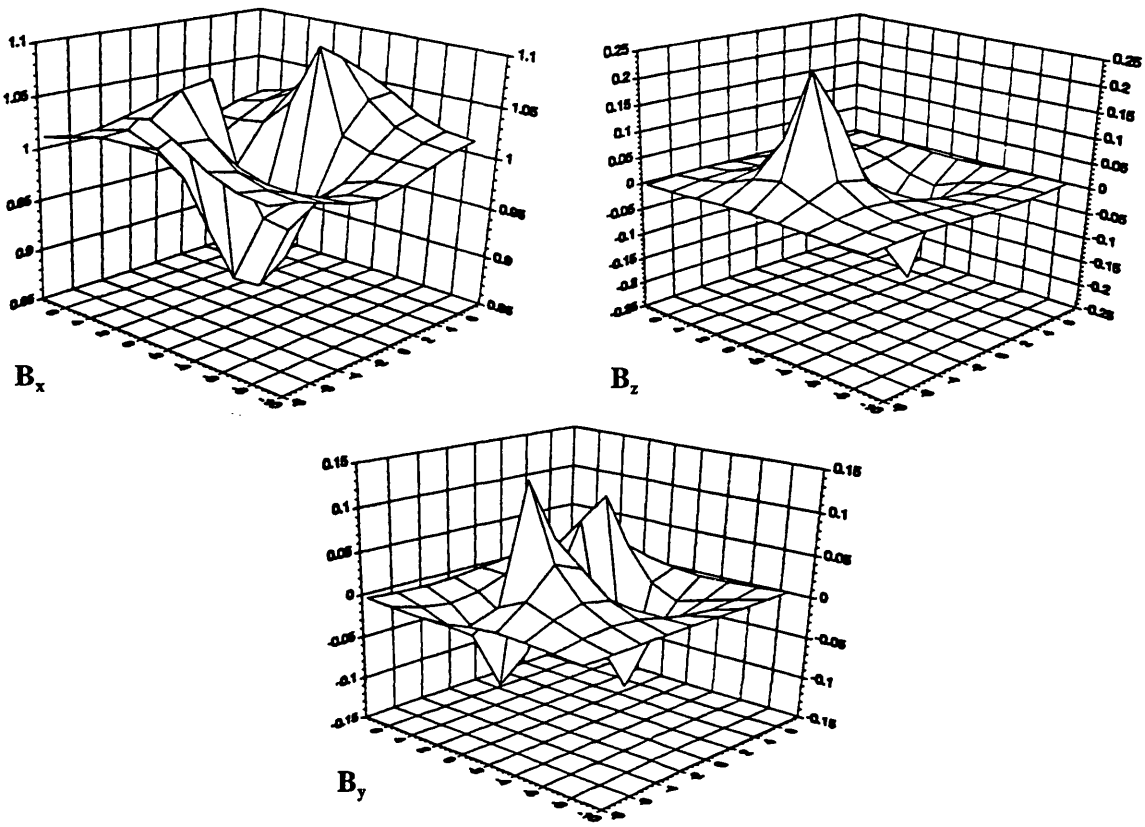


Figure 7.1 The magnetic field **B** *over a crack (horizontal plane corresponds to the surface of the component, the crack lies in centre in direction from front left to back right)*

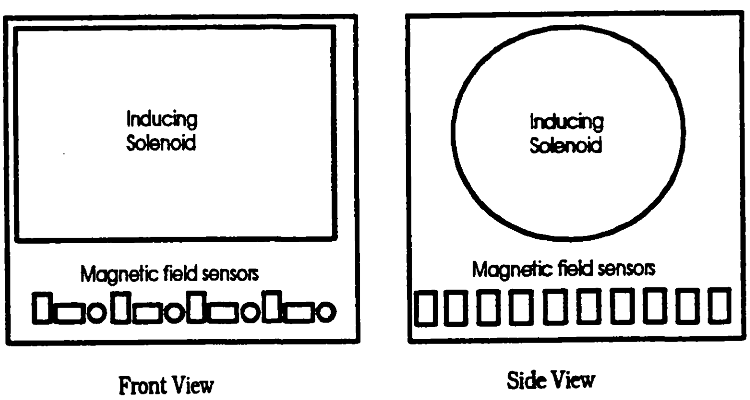


Figure 7.2 Schematic diagram of the ACFM probe showing arrangement of coils

The solenoidally induced field is not uniform. To apply a first-order correction to this non-linear phenomenon, the probe is used to scan a flat featureless surface and this background reading subtracted from scans for investigation. This procedure will also offer some correction for irregularities and noise in the magnetic sensor coils. Use of more than one solenoid, in an appropriate geometric formation, can make the induced field more uniform (Zhou et al, 1993).

7.1.2 The Architecture and Requirements of the AIRES KBS System

The AIRES project used electromagnetic and vision inspection techniques to detect, classify, and characterise defects in machined metal components. Defects must first be identified and then classified and characterised automatically, by initially making use of expert knowledge about the inspection sensor employed and then by combining the results of both.

One way of automating the interpretation of inspection data is to produce software which imitates the procedures and reasoning used by a human expert inspector. For this the software will need to store and manipulate information related to:

- the component being inspected
- the sensor employed and its mode of operation
- sensor data interpretation procedures
- comparison of different inspection technique data or data fusion

One type of software architecture which is particularly suitable for the interpretation of multi-sensor data is that of the blackboard architecture.

7.1.2.1 <u>Architecture of the Knowledge Base System</u>

The electromagnetic and vision sensor data analysis routines were incorporated within a architecture based on blackboard system concepts. The architecture of the knowledge base system is illustrated in Figure 7.3.

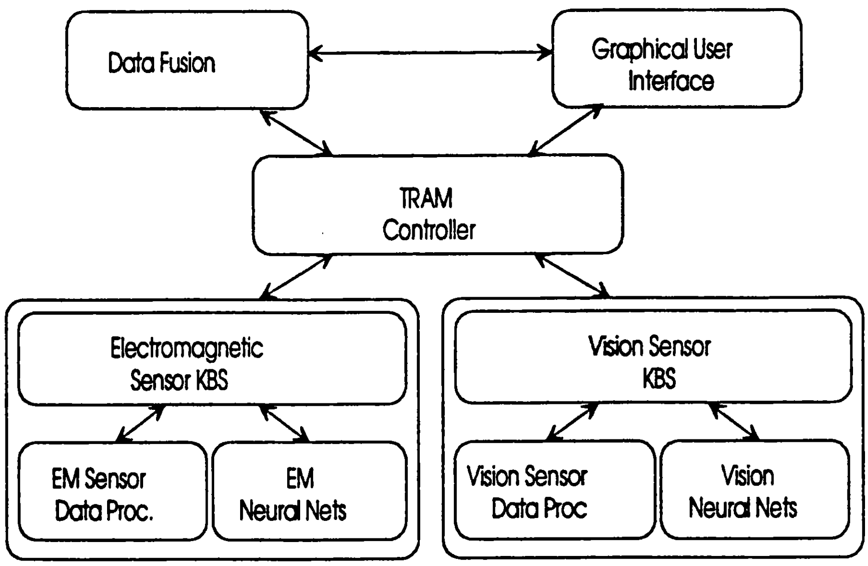


Figure 7.3 Knowledge base system structure for AIRES

The blackboard architecture is described by Feigenbaum, in Englemore and Morgan (1988), as the most general and flexible knowledge-based architecture. In particular, it allows several levels of sub-systems to interact. The main sub-systems here are the electromagnetic sub-system including neural networks (Electromagnetic Sensor KBS), the vision sub-system and neural networks (Vision Sensor KBS), the Data Fusion subsystem and the Graphical User Interface. An outline of the basic requirements and functions of the electromagnetic, vision, data fusion and of the user interface are described below.

7.1.2.2 User Interface

There is a man-machine interface (MMI) that is a text based user interface. For the end-user, an additional graphical man-machine interface for the final AIRES system was required to show the results of the reasoning and interpretation carried out. This interface would allow user interaction via menus, dialogue boxes, etc., and with graphical display of results. A suitable form of results display is that of a picture of the component in question with detected and characterised defects shown graphically on the surface of the component. A basic design, as in Figure 7.4, for a Windows environment was implemented.

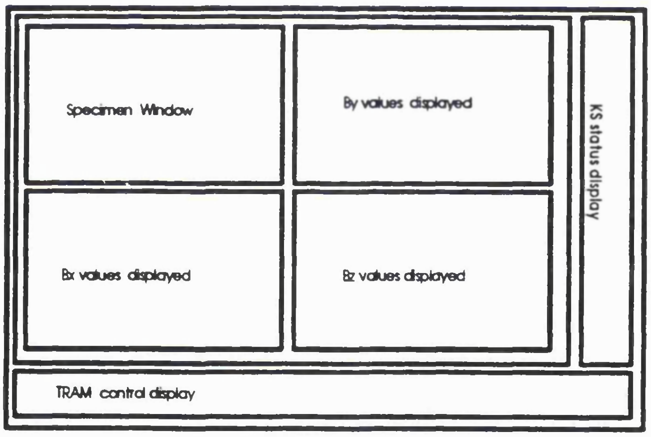


Figure 7.4 Graphical user interface display

7.1.2.3 The Vision Sensor

The vision system in AIRES makes use of standard image processing methods described by Chin (1982 and 1988) for the detection of surface defects in fixed, and usually optimised, environments, with intelligent control to select the best available method for a particular situation. These routines are also augmented by the use of vision neural nets. The vision sensor consists of a monochrome CCD camera, with computer controlled focus and lighting. A toolkit of image processing operations was developed for dedicated image processing hardware. The image processing operations are available to a number of knowledge sources, which together comprise the vision

KBS. This work was carried out independently by other groups at University College London and in Germany at the University of Hanover and Deustche Aerospace.

7.1.2.4 The Electromagnetic Sensor

The electromagnetic inspection technique applied here is based on the Alternating Current Field Measurement (ACFM) method, which works by inducing a high frequency alternating current in the component inspection surface and studying the resulting magnetic field above the surface. The data is collected in the form of a three matrices of **B** values, one matrix for each direction and each point in the matrix for a sample location. The 3-D surface of the magnetic field can be analysed for signs of defects, for their type (pit, crack, or other) and their parameters. The algorithms used were based on the theoretical modelling of ACFM. A set of electromagnetic neural net routines were also used to analyse the magnetic data as a complement to the ACFM theoretical algorithms.

7.1.2.5 Data Fusion

Combining the results of two sensors will in general give a more robust and powerful system. In this case, the combination of visual data and the electromagnetic data can work well. Both the electromagnetic and vision techniques can detect surface defects only. The ACFM technique works through coatings and can, given good conditions and particular defect geometries, give accurate readings of the defect sizes, in particular for cracks. In contrast, the visual technique is somewhat more robust in terms of required conditions for initial detection of defects, but cannot give depth measurements, although gives better results for pits. A fuller interpretation of the nature of the object is obtained by combining electromagnetic field and visual information. Making use of the vision data early on would provide more accurate initial interpretations of the electromagnetic data, although it is only data from the electromagnetic sensor which will indicate the depth of defects such as cracks. Consequently, a data fusion module is required to merge the interpreted results from the electromagnetic signals and the vision data.

7.1.3 The Application and Phases of Work

The AIRES multi-sensor system was to be used for inspection of metallic components, since the electromagnetic principle does not function on non-metallic surfaces. The approach is intended to be applicable to on-line automatic inspection tasks of components in production in the automotive and aerospace industries.

The requirements for the AIRES system was that it should be able to inspect:

1 Standard plate specimens with features such as pits, cracks, high spots, slots etc.

2 A camshaft to examine surface defects on ground faces.

The first geometry was chosen as a test specimen for the initial work. The second is meant to represent the type of component, that might be found in a real application.

The work took place in two phases. In the first phase, a prototype inspection system was developed to inspect flat plates, with limited set of defects and free of corrosion. Each sensor in this phase was to complete a scan before interpretation of the data and to act independently of each other with manual control. Theoretically generated data was used initially, until real sensor data was gathered. In the second phase, the aim was to have a system which could inspect a realistic component with more realistic defects. In addition, a more integrated and opportunistic use of the sensors was investigated, with more sophisticated computer control.

7.2 THE BLACKBOARD SYSTEM

Blackboard systems have been in use for about twenty years with varying degrees of success in many different applications. There are several reasons for a blackboard approach to software development that can be applied to supporting a blackboard system for the AIRES application (Corkill, 1991). Firstly, the AIRES project involved several development groups, and the modularity and independence of the blackboard KS structure allows for separate development and testing of code. Moreover, AIRES includes two main separate areas of expertise, electromagnetic and vision, each requiring its own knowledge representation: a blackboard system allows for easy incorporation and combination of these two differing approaches. The arguments for using a blackboard approach on a project diverse both in location and substance as this are compelling. Each subsystem could be developed independently and combined at later stages of the project. The only requirement was for the blackboard data structures to be defined early on in order to be able to communicate across the two subsystems.

There are other benefits from the use of blackboard systems. They provide

- dynamic control, allowing opportunistic reasoning and an incremental approach to problem solving
- multilevel data to allow reasoning based on data of various levels of granularity

These facilities allow flexibility of reasoning and control for AIRES. The first feature allows information from electromagnetic and vision sensors to be built up incrementally by combining sensor results and re-scanning where needed. The second feature allows the use of a range of computational processes, from numerical procedures which act on raw sensor data to high-level knowledge to be applied to reasoning based on characterised defect data.

7.2.1 The Development Tool, TRAM

The AIRES system was built using TRAM, which was originally developed for autonomous mobile robotics (Tableau noir pour la Robotique Autonome Mobile) as a multi-expert system development tool based on a blackboard architecture (Koenig & Crochon, 1989; Koenig et al 1990).

TRAM has a typical blackboard architecture (Nii, 1986a and b); Engelmore & Morgan, 1988) with knowledge sources that are independent modules accessing a common database structure, known as the blackboard, which is composed of several abstraction levels, and a the controller, or inference engine, which manages the knowledge sources and the blackboard. As described in detail in Section 4.1.2, the blackboard constitutes the system data structure. The knowledge sources operate on the data on the blackboard. The execution of the knowledge sources is controlled by the system controller. When the blackboard attains a certain state, a KS may become operative, and this KS will update the blackboard, which in turn may trigger further KS operation.

The structure of TRAM under Unix is illustrated in Figure 7.5.

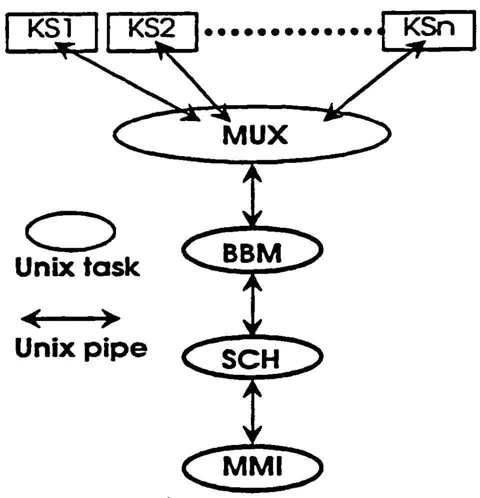


Figure 7.5 The structure of TRAM

The core of TRAM is made up of the processes BBM and SCH. BBM is the Blackboard Manager, responsible for access to the blackboard databases and for establishing which KSs may be activated. SCH is the Scheduler, which selects the executable KS to carry out next. TRAM was written in C, and hence the knowledge sources and concepts refer to C structures and functions. Procedures in other languages can also be integrated by using a C function interface. In addition, MMI, TRAM's text-based user interface man-machine interface, is used to input operator commands and provides the operator with details of the function of TRAM, and MUX multiplexes and de-multiplexes the communication between KS and BBM.

7.2.2 Blackboard Concepts

The blackboard can be considered as an object-oriented database composed of a hierarchy of concepts. Automatic interpretation of inspection data requires that the software make use of information on:

- the component being inspected
- the sensors employed and their mode of operation
- initial sensor data
- interpretations of sensor data
- data fusion results

Thus the concepts defined for this application have to represent each one of the above. In the TRAM tool, each concept has structure shown in Table 7.1.

Table 7.1 TRAM concept structure

concept name	the name of the TRAM concept
main structure	a C structure type, accessible only by a KS
inherent attributes	C structure types accessible by both KSs and the BBM
attached attributes	C structure types accessible only by the BBM
relation attributes	pointers to other concepts in the blackboard

The structure of TRAM concepts reflects frames, described in Section 4.1.2, in that the relation attributes indicate the relationship between classes in the blackboard, and inherent attributes are equivalent to slots in a frame or concept. Attached attributes and main structure have no direct equivalence in traditional frame-based systems, but are defined to provide extra efficiency in that C data structures do not have to be coded for the attached attributes, since they will only be used for reasoning by the BBM and high level structures do not have to be coded for the data which is to be passed to C functions.

The knowledge representation in this work reflects the findings of previous work in general sensor fusion that three levels of data or information are stored (Pau, 1989). The first or lowest level here corresponds to the pre-processed sensor data, the mid level stores data about the existence of features, and finally the top level is that of the characterised defects. The classification and characterisation routines used classical signal processing algorithms and neural network models.

7.2.3 Knowledge Sources

The core of a KS in TRAM is a C function which carries out numerical or other routines. A mechanism is then required to link the C function to the blackboard, to feed data from the blackboard to the function and to make use of the output from the function to update the contents of the blackboard. This is fulfilled by the knowledge source card which provides TRAM with a description of when to use the C function, how to provide the function with data and the effect that the output from the function will have on the common data storage or blackboard.

KS cards have the structure given in Table 7.2.

Table 7.2 TRAM knowledge source card structure

program	the name of the KS C function
activation condition	when the blackboard database satisfies the specified condition, the KS is activated
input parameters	the KS C function input parameters
proposition	provides instructions for the BBM to update the blackboard with the KS results
output	the KS C function output parameters
error test	provides a condition to activate error handling
error action	specifies the particular error action

The activation condition has a relatively easy-to-read grammar. Error actions provide a way for the BBM to detect problems and rectify them.

7.2.4 The Agenda

In large applications, reliance on full opportunistic reasoning may be inappropriate. Making the BBM check all KSs in case they can be triggered can be very inefficient and can slow the process considerably. To overcome this problem, TRAM allows the use of an agenda that indicates which KSs may be triggered at each stage of the process. This agenda encapsulates the knowledge about the best strategy for making use of the KSs and is based on flowcharts describing the complete process of operating the sensor and interpreting the data.

7.2.5 Interpretation of the Electromagnetic Sensor Data

Within AIRES, the electromagnetic sensor data needs to be processed and the results interpreted. In order to do this, a full understanding of the form that the data takes and of how the sensor may be utilised to obtain data is required. This section discusses the algorithms designed to process and interpret the signals from the electromagnetic sensor.

The ACFM technique can detect surface defects in metallic components. The classes of defects that can be detected by ACFM can be loosely defined as cracks or notches, pits or areas of changed electromagnetic properties. The electromagnetic sensor outputs matrices giving the *x, y,* and *z* components of the magnetic field **B** over the surface. This data can be interpreted to give the geometry of the defect and the size. The modelling theory, however, is only fully developed for hemispherical pits and cracks or notches which are semi-elliptical in shape. So for general defect shapes, classification and characterisation can only be approximate.

The precise numerical form of the magnetic field above a crack can be accurately predicted, and by comparing the experimentally recorded data to the predicted values, the crack dimensions may be established (Michael et al, 1991). A similar method may be used to detect and size any other defect, such as a pit, for which the numerical magnetic field can be predicted. The practical use of the numerical solutions requires these to be implemented as look-up tables. The use of these tables requires a defect to be identified and its alignment given. This procedure is now explained.

7.2.5.1 Typical Theoretical electromagnetic Data

From Figure 7.1 on page 261, where a clear pattern in the magnetic field can be seen, it appears that a human expert would be able to classify or at least detect a defect. These patterns can be described in terms of *peaks* and *troughs*. Hence the work carried out initially concentrated on classification based on attempting to encapsulate the reasoning that an expert would carry out for classification.

The **B** surfaces displayed in Figure 7.1 each have a characteristic shape. The important characteristics and use of each are given below.

$\mathbf{B}_x$ **Two peaks and a trough lying in between.** The length of a crack is related to the distance between its $\mathbf{B}_x$ peaks; an exact analytical relationship between the two may be calculated using electromagnetic theory. The depth of a crack or other defect geometries may be also be calculated from similar quantities extracted from the $\mathbf{B}_x$ peak and trough co-ordinates.

$\mathbf{B}_y$ **Two peaks and two troughs.** The relative positions of these are used to distinguish between pits and cracks.

B_z **One peak and one trough.** The B_z peak and trough co-ordinates and values may be used for a similar purpose to the B_x ones, and are used as supplementary data.

7.2.5.2 Typical Classification of Defects

The ACFM method, as incorporated in AIRES, works as follows for the inspection of a defect on an otherwise smooth surface:

1. For each of the B_x, B_y and B_z data sets, the co-ordinates and the values of the peaks and troughs are found.

2. The feature is then classified as either a pit or a crack or of unknown type, by using the four B_y turning points (see Figure 7.6):

   - If the angle α between the lines joining opposite pairs of turning points is small enough, the defect is a crack.
   - If α is large enough, the defect is a pit.
   - The value of α outside given ranges will signify a unknown defect type.

3. If a crack or pit exists, the length and depth are then calculated by using the co-ordinates and values of the B_x and B_z peaks and troughs and a set of look-up tables which relate these dimensions to the actual crack dimensions. Similarly for a pit.

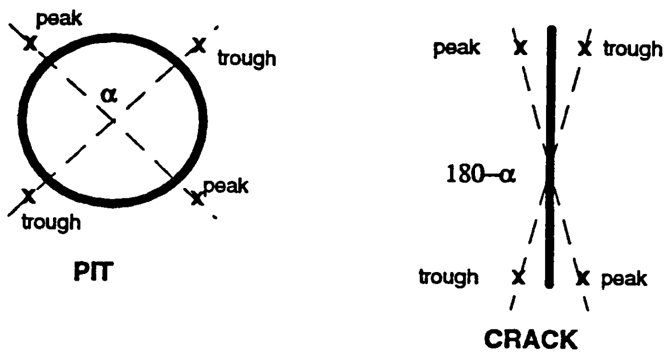


Figure 7.6 $\mathbf{B}_y$ *peaks and troughs for cracks and pits*

7.2.6 The Signal Interpretation Procedure

As in Figure 7.7, the electromagnetic sensor KBS carries out signal interpretation in four steps:

7.2.6.1 Pre-processing

Collected sensor data is pre-processed to give to the electromagnetic sensor KBS data free of noise and irregularities due to sensor sensitivity.

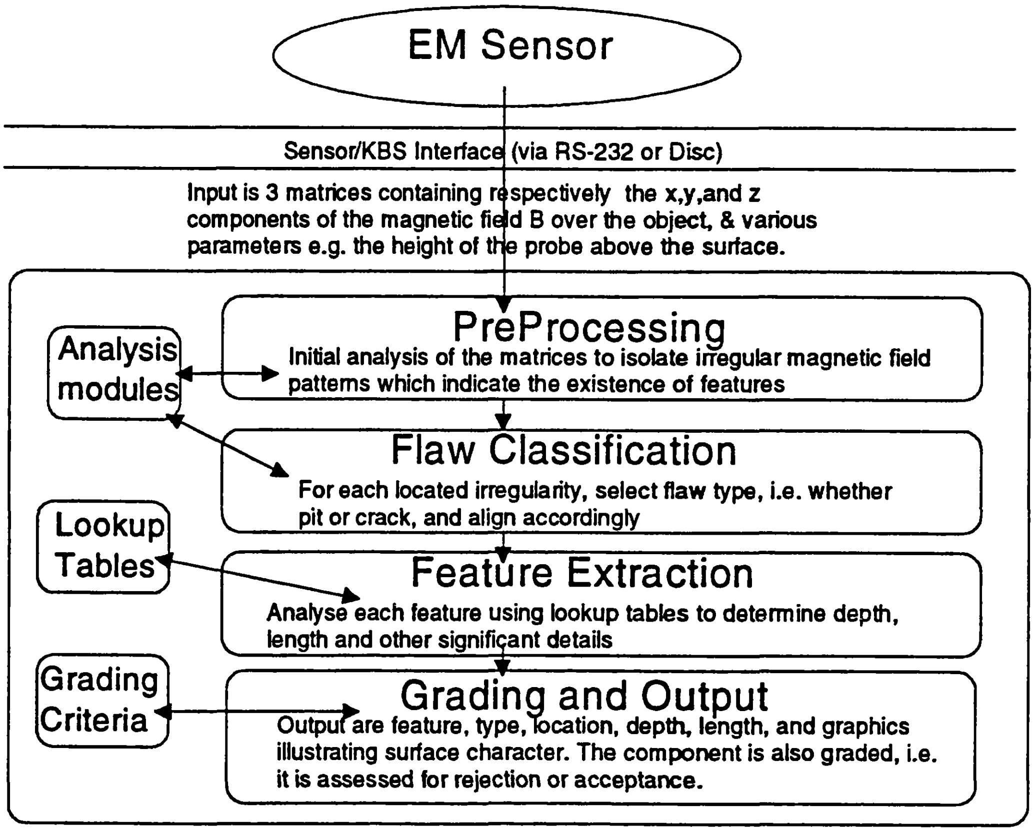


Figure 7.7 Electromagnetic sensor KBS signal interpretation procedure

7.2.6.2 Flaw Detection

A flat featureless surface will produce a uniform magnetic field. Any irregularity in the magnetic field indicates the existence of surface features. During the pre-processing an initial analysis of the three **B**-matrices to isolate portions corresponding to features or flaws is performed.

It was found that flaw detection was best performed by a neural network; a function based on the electromagnetic theory was found to be very much less effective and efficient. Thus the Phase II system contained only one KS to carry out this step which was based on a neural network developed in parallel to the analytical theory based KSs by the group at Deutsche Aerospace.

7.2.6.3 Flaw Classification

Each type of surface flaw produces different magnetic fields; the nature and shape of the magnetic fields arising from pits, cracks, and high spots have been described and analysed. At this stage a given flaw or feature which has been spotted by the pre-processing is classified. A feature is defined as one of the following: pit, high spot, slot, pore, inclusion, or crack. In the developed demonstrator electromagnetic KBS, only features of the type pit, crack and unknown are considered.

A flaw such as a crack has a direction and this may not be parallel to the edge of the scan, but the feature detection procedure assumes that it is. Thus data has to be aligned in the crack direction, which effectively means rotation of the x and y co-ordinates. Re-alignment is carried out, not by physically moving the probe, but by a mathematical transformation of the data.

7.2.6.4 Feature Extraction

The next stage is to extract the characteristics of each feature, e.g. the depth and length for a crack. Lookup tables are provided which relate the height z of the electromagnetic probe above the surface and certain values extracted from the magnetic field matrices to the length and depth of the crack; similar tables are provided for other features.

7.2.6.5 Grading/Output

Finally the surface details are output and the surface is evaluated using the grading criteria.

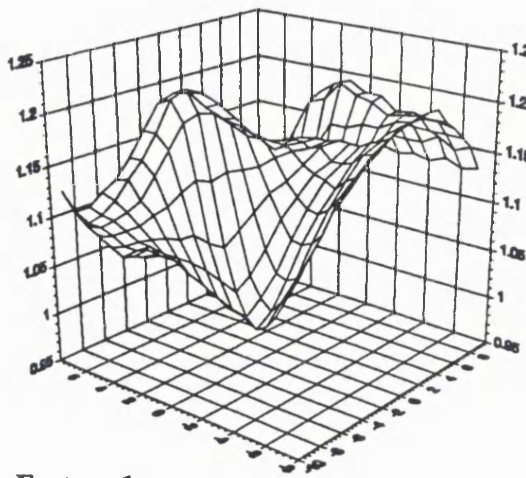
7.2.7 Issues in Real ACFM Sensor Data Interpretation

The actual data produced by a sensor reproduces the theoretically predicted data with varying fidelity. Consider the $\mathbf{B}_x$ data shown in Figure 7.8(a). In the scan of feature 1, the characteristic shape of the $\mathbf{B}_x$ surface may be seen: two peaks with a trough in between, although the trough is almost completely obscured in the figure. In the second scan, this shape is hardly visible and it appears to be a featureless irregular surface. It would be hoped that AIRES would provide a reasonably accurate analysis of the first data, but not so for the second set of data. Note that the first data set was obtained from scanning a feature with five times the linear dimensions of those of the second data set. The sensor will have a limited resolution depending on the size of the sensor coils, their sensitivity, and other parameters.

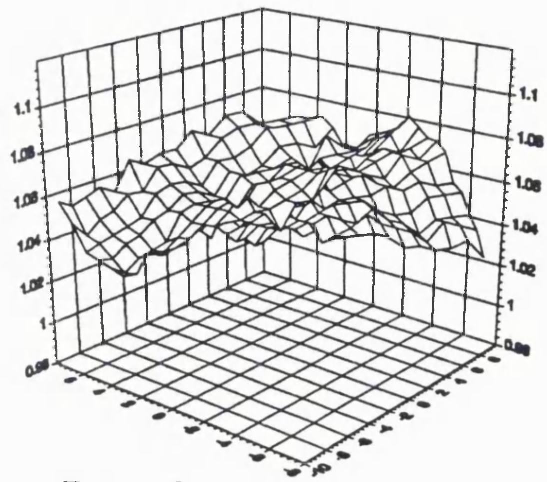
As already discussed, the key to the ACFM analysis is the identification of the peaks and troughs of the data. Once these are known, the simplistic rules relating the relative locations of the peaks can be applied for classification. Although this is relatively easily done by a human expert, identifying peaks and troughs automatically is essentially a problem of surface analysis. As can be seen from Figure 7.8(a), the likelihood of doing this, and the usefulness of any solution, depend significantly on the data set $\mathbf{B}$.

Two issues concerning real ACFM sensor data are here considered. The first issue is related to removal of noise introduced into the sensor signal. The second and, from a computational point of view, more complex problem is the identification of the sensor signal pattern, that is of peaks and troughs, for initial classification and further characterisation.

(a)Before masking:

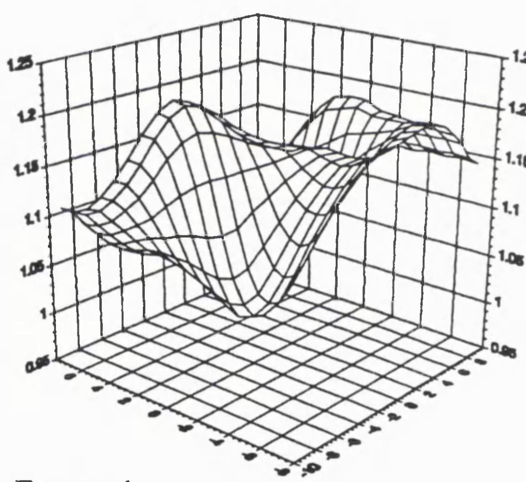


Feature 1

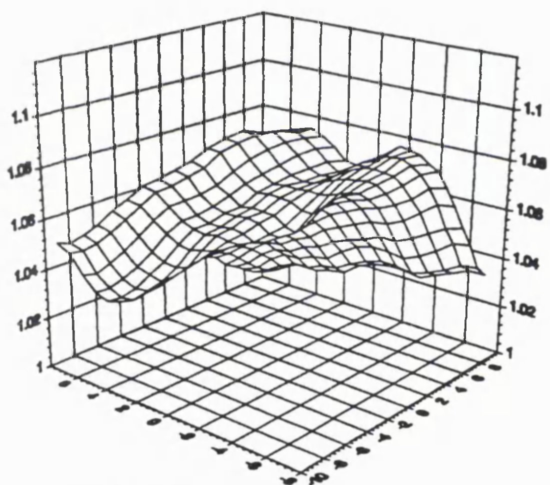


Feature 2

(b)After masking:



Feature 1



Feature 2

Feature 1 is a notch 10mm long and 5mm deep. Feature 2 is a notch 2mm long and 1mm deep. Both data sets have been corrected for solenoidal irregularities and noise. The unit of the x and y axes is 1mm.

Figure 7.8 $\mathbf{B}_x$ *data for two features*

7.2.7.1 Removal of noise

The operation of the electromagnetic sensor is very delicate as the electromagnetic sensor is highly sensitive to small deviations in positioning. Real data can be distorted by noise usually caused by the geometry of the component and by any slight defect in use.

It was found that for the purposes of the initial system, simple masking was sufficient to remove some of the effects of noise. The masked data allowed the underlying peaks and troughs to be more easily identified.

Figure 7.8(b) shows the data of Figure 7.8(a) after masking. A simple masking formula has been used with no weighting:

$$\text{Masked }(B_{m,n}) = \frac{1}{9} \sum_{i=m-1}^{m+1} \sum_{j=n-1}^{n+1} B_{ij} \qquad (7.1)$$

where $B_{i,j}$ denotes the value of the relevant **B** component at a point $P_{i,j}$, one in a mesh of points as shown in Figure 7.9. The formula is easily modified for corner points and edges.

$P_{m-2,n-2}$	$P_{m-2,n-1}$	$P_{m-2,n}$	$P_{m-2,n-1}$	$P_{m-2,n+2}$
$P_{m-1,n-2}$	$P_{m-1,n-1}$	$P_{m-1,n}$	$P_{m-1,n-1}$	$P_{m-1,n+2}$
$P_{m,n-2}$	$P_{m,n-1}$	$P_{m,n}$	$P_{m,n-1}$	$P_{m,n+2}$
$P_{m+1,n-2}$	$P_{m+1,n-1}$	$P_{m+1,n}$	$P_{m+1,n+1}$	$P_{m+1,n+2}$
$P_{m+2,n-2}$	$P_{m+2,n-1}$	$P_{m+2,n}$	$P_{m+2,n+1}$	$P_{m+2,n+2}$

Figure 7.9 Mesh of points

The first data displayed in Figure 7.8(b) shows only a slight smoothing; but the second is massively smoothed. If an unmasked peak/trough lies at $P_{m,n}$, the location of the corresponding masked peak/trough will be at one of the eight adjacent points or $P_{m,n}$ itself, which enables the unmasked location to be found once the masked location is known. Masking may reduce the value of maxima but the value of a maximum before masking may be obtained by reference to the original unmasked data and similarly for the minima.

For more complex problems of noise, further work may require the application of more complex algorithms from signal processing techniques.

7.2.7.2 Peak/Trough Location Methods

Three methods have been considered for locating the peaks and troughs of the masked **B** data.

- **Simple Searching for Peaks and Troughs**

In this method, a search is carried out through all the data points for the greatest and least values. This method is naive and cannot be relied upon for any but the most regular surfaces. For example, if in a set of $\mathbf{B}_x$ data, where two peaks are expected, one peak P_1 has height h_1, the other (P_2) height h_2, with $h_1 > h_2$, one of the adjacent surface points to P_1 has height h with $h_1 > h > h_2$, and h_1 and h are the greatest $\mathbf{B}_x$ values in the surface, then such an algorithm will record h_1 and h as the two greatest $\mathbf{B}_x$ values and hence miss the peak P_2.

- **Location of Local Maxima and Minima**

Here, a search is carried out to locate groups of data points which make up peaks and troughs. This second method is more robust than the simple search and works in most situations. Criteria used to check for peaks and troughs are that a maximum value is greater than the surface values at all eight adjacent points and, for robustness, it is greater than each adjacent point of a maximum has value greater than the values at all adjacent points to it lying further away from the maximum and likewise for a minimum. In detail, consider a point $P_{m,n}$ surrounded by a mesh of points, as in Figure 7.9. The criteria used in the algorithm for $P_{m,n}$ to be a maximum are:

1. $P_{m,n} \geq P_{m+i,n+j}$ for i=-1,0,1; j=-1,0,1

2. The value at each diagonal point adjacent to $P_{m,n}$ must be greater than or equal to the values at the three neighbouring points which are furthest away from $P_{m,n}$; thus $P_{m+1,n+1} \geq P_{m+1,n+2}$, $P_{m+1,n+1} \geq P_{m+2,n+1}$, $P_{m+1,n+1} \geq P_{m+2,n+2}$ and similarly for $P_{m+1,n-1}$, $P_{m-1,n-1}$, $P_{m-1,n+1}$

3. The value at each non-diagonal point adjacent to $P_{m,n}$ must be greater than or equal to the values at the three outward adjacent points; thus $P_{m+1,n} \geq P_{m+2,n+1}$, $P_{m+1,n} \geq P_{m+2,n}$, $P_{m+1,n} \geq P_{m+2,n-1}$ and similarly for the other points $P_{m,n+1}$, $P_{m-1,n}$, $P_{m,n-1}$.

To search for a minimum, "less than or equal to" is substituted for "greater than or equal to".

The algorithm is applied as a two-test process. The first test finds and records all points satisfying condition 1. Then these points are tested for 2 and 3. The procedure is easily altered if $P_{m,n}$ is an edge or corner point, or if it is adjacent to an edge point, to exclude non-existent points. This method will locate all local maxima and minima, and this of course may still provide inconclusive results owing to data irregularity, as too many peaks and troughs may be given.

For example, applying this algorithm to the first set of data in Figure 7.8(b), four local maxima and three local minima are found to exist. One minimum and one maximum lie at the corners, and may therefore be omitted from further consideration, but that still leaves three local maxima and two local minima. To decide which two maxima and one minimum to consider, various rules may be used. For example, the $\mathbf{B}_x$ peaks and troughs should lie along the same line as the $\mathbf{B}_z$ peaks and troughs, and if the $\mathbf{B}_z$ peaks and troughs can be located, their positions can be used to decide upon the true $\mathbf{B}_x$ peaks and troughs.

This method was chosen to be applied in AIRES.

- **Fitting Surfaces to the Magnetic Field Data**

This method involves fitting the **B** surfaces with surfaces of known equations, whose peaks and troughs are known and thus model known signals from defects, by least squares minimisation or other numerical techniques.

The difficulty with this method is to find appropriate analytical functions to map the $\mathbf{B}_x$, $\mathbf{B}_y$ and $\mathbf{B}_z$ surfaces. Work was carried out to find possible candidates. Unfortunately, although reasonable exact-fitting functions were found for some theoretically generated surfaces, the same functions do not map the surfaces derived from actual sensor scans. Further work could be carried out in this area by considering advanced methods for signal interpretation, such as wavelets which can be used in pattern matching (Mallet, 1989; Wu & Du, 1996).

7.3 Knowledge Sources and Concepts for the Electromagnetic Sensor

The purpose of the knowledge elicitation process for the electromagnetic sub-system was to encapsulate the expertise required for the interpretation and classification of electromagnetic data, using the ACFM techniques, as a set of concepts, knowledge sources to act on the concepts and an agenda for making efficient use of the knowledge sources.

The concepts representing the defects are based on the classes of defects that can be detected by ACFM, which are cracks or notches, pits, and general areas of changed electromagnetic properties. The electromagnetic signal interpretation can be approximately divided into the following steps:

1. Scan the component and collect the data.
2. Carry out any pre-processing which may be required.
3. Decide if a feature is present in the given signal data.
4. If a feature is present then classify it as one of: crack, pit or unknown defect, or possibly other.
5. For a defect which has been successfully classified, characterisation then takes place.

6. If the feature has not been satisfactorily classified then re-scanning may take place based on the results of data fusion.

The electromagnetic sensor outputs matrices of the x, y, and z components of the magnetic field **B** and these are interpreted to give the geometry of the defect and the size. For general defect shapes, the classification and characterisation given can only be approximate at this stage, since the modelling theory is only fully developed for cracks or notches which are semi-elliptical in shape and for hemispherical pits.

It was envisaged that knowledge sources based on the mathematical modelling of the field would be sufficient. It was found nonetheless that much work was required to provide sufficiently smooth data before the algorithms designed could be applied. Even more surprisingly it was found that simple neural networks created independently of the electromagnetic theory based knowledge sources by others were found to be very successful at flaw detection.

7.3.1 Detailed Design of Concepts and Knowledge Sources

As described above, four groups of knowledge sources were required for this sub-system:

- Electromagnetic Sensor Control
- Electromagnetic Signal Interpretation functions
- Data Fusion procedures
- User interface input functions

The following outlines the concepts and gives example KS cards. Most KSs have an associated C function which has the same name as the KS itself. All KS names begin with the prefix EM, for example EMInspect and so on, to indicate that they are part of the EM KBS as opposed to the vision and main KBS. Neural network KSs for the electromagnetic sensor KBS have function names beginning with EM_NN.

In the AIRES system, the KS Card entries for the error test and the error action are exactly the same for all KSs. The test is carried out on the parameter *errorcode*, an output variable from the C function which is additional to the ones given explicitly in the following descriptions. If an non-zero errorcode is returned by the C function, then an EMError object is posted on the blackboard with the attribute *state* set to the 'error_has_occurred' and *code* set to the errorcode value. In the prototype AIRES system, the system would halt and output the error information to the user. In future extensions, it is foreseen that the errorcode can be used to identify the procedure required to recover from the error.

7.3.2 Electromagnetic Sensor KBS Concepts

Only the concepts required for the electromagnetic sensor KBS are considered here. These are concepts which represent the object being inspected, the electromagnetic sensor, the electromagnetic sensor data and interpretations of the electromagnetic sensor data and finally aspects of data fusion control.

Figure 7.10 below illustrates the set of concepts implemented for the Phase II electromagnetic sensor KBS. The arrows indicate relational attributes, so for example, the concept EM3dData is linked to the concept EMSensor. The three lower boxes contain the main structure, inherent and/or attached attributes, and relation attributes, in that order.

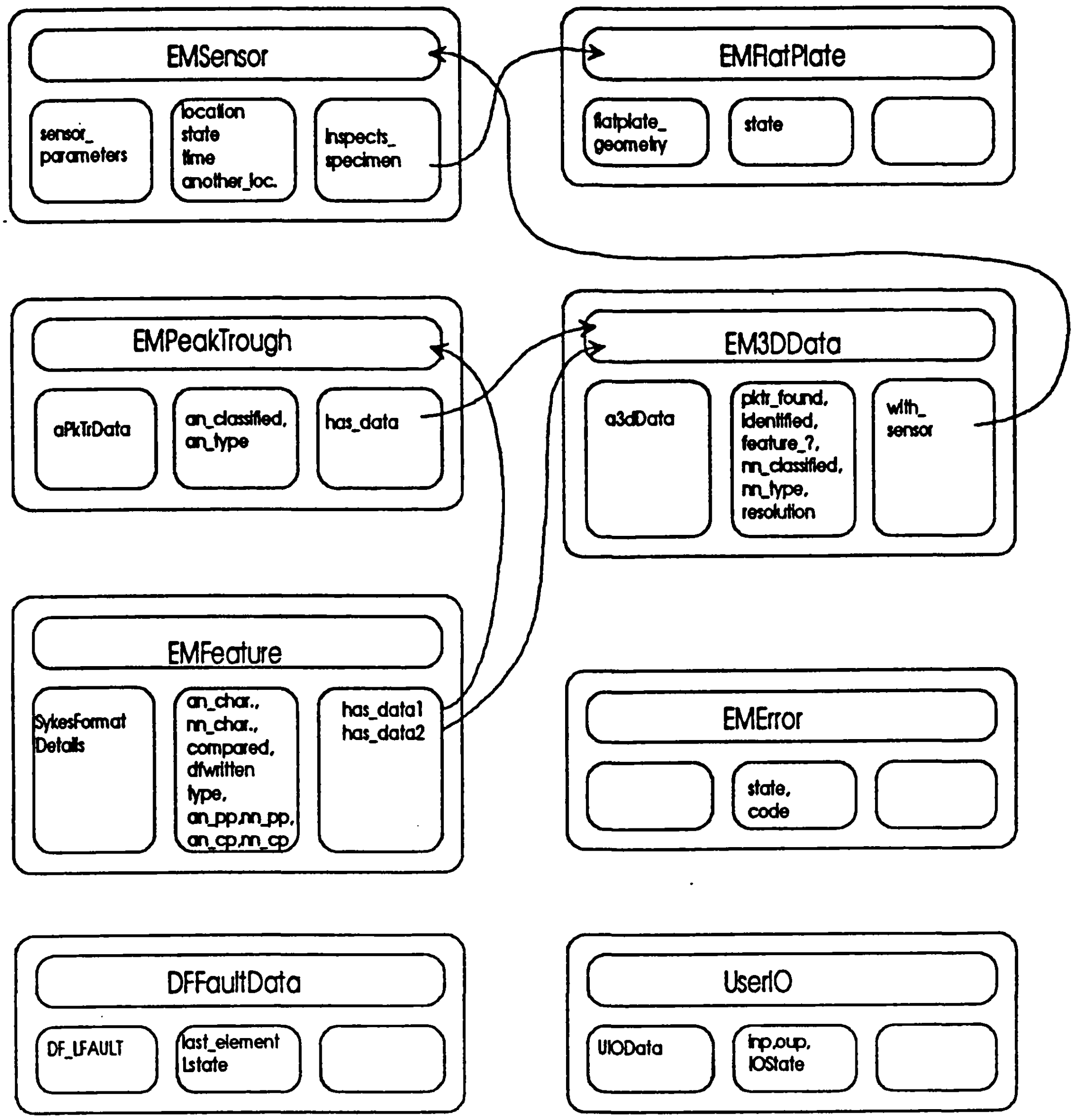


Figure 7.10 Electromagnetic sensor KBS concepts

When a component, which here was limited to a flat plate or camshaft, is presented to the system for inspection, an instance of one of the EMFlatPlate or EM Camshaft concepts is created on the blackboard. This instance stores all the necessary data pertaining to the component. For the inspected object, it was found that the concept for a flat plate required only minor modifications in order to model the camshaft component. By using polar co-ordinates to map the surface of approximately cylindrical components onto a flat surface, the same attributes could be used for the EMCamshaft concept as for the EMFlatPlate concept.

The AIRES system starts by setting up the electromagnetic sensor (and vision) sensor(s) to inspect the plate, and details of the sensor, such as its location, its lift-off from the surface, etc., are stored in an instance of EMSensor. Every time a set of **B** data is received by the electromagnetic sensor KBS from the sensor, it is stored in the blackboard as EM3dData objects. Two EM3dData instances are created at one time: one is at a low resolution for use by neural nets for identification of possible features and for classification and characterisation, the second is used during characterisation by analytical methods.

In the initial prototype developed during Phase I, each set of **B** data was analysed to find the peaks and troughs needed for classification and characterisation, and the details of these peaks and troughs are stored in an instance of EMPeakTrough. This object was then inspected to decide on a classification. In the final prototype developed during Phase II, this was replaced by only one neural net KS that classified flaws in one step.

The existence of flaws in the electromagnetic sensor data is indicated by instances of EMFeature. This object has as an attribute the location of the feature within the **B** data. Once the feature has been classified and categorised, the results are also stored in this concept, either using crack or pit related attributes.

A common set of KBS concepts for the output of sensor interpretation results was specified to facilitate the data fusion process: these were DFFaultData and DFControl. This resulted in a change for the electromagnetic sensor KBS concepts: the two concepts EMCrack and EMPit were combined into one EM Feature concept, which was also used to give an initial indication of the existence of a possible feature.

General concepts were used to communicate with external sources to the Electromagnetic KBS:

- The EMError concept which is used to indicate errors occurring when executing a knowledge source. It records any error code output by the relevant treatment function.
- DFFaultData is a data fusion concept common to both the Vision and the Electromagnetic Sensor KBS to allow interKBS communication.

- A concept which would trigger output to the users on the screen was also defined called UserIO.

Two example concepts are given in Table 7.3.

Table 7.3 Example concepts

concept name	**EMSensor**
main structure	SENSORPARAMETERS
inherent attributes	state: SENSORSTATE, location: POSITION, time: EMTIME, another_location: YN
attached attributes	
relation attributes	inspects_specimen: EMFlatPlate
concept name	**EM3dData**
main structure	a3dData
inherent attributes	pktr_found, identified, feature_exists, nn_classified: YN, nn_type: FEATURETYPE, resolution: high, low
attached attributes	
relation attributes	with_sensor: EMsensor

7.3.3 Overview of Knowledge Sources

The above short description of the use of the electromagnetic sensor KBS concepts needs to be supplemented by explaining the electromagnetic sensor KBS knowledge sources and their operation.

Figure 7.11 shows a flowchart of the KS without reference to the blackboard. Figure 7.12 shows the interaction of the KS and the blackboard. From these figures it can be seen that the electromagnetic sensor KBS starts the operation with EMInit which creates instances of the EMFlatPlate and EMSensor in the blackboard and assigning values to their attributes.

When an instance of EMSensor exists in a certain ready state, that is, all appropriate parameters have been initialised, it triggers the operation of EMScanning. This KS controls the interaction between the sensor and the KBS: it sets the sensor position and lift-off from the plate surface and then reads in the magnetic field data from the sensor and this is stored in an instance of EM3dData.

A complete instance of EM3dData triggers the EM_NNIndicate KS. If a feature is indicated in the data, then an instance of EMFeature is created. Only if an instance of EMFeature exists will the classification and characterisation process start.

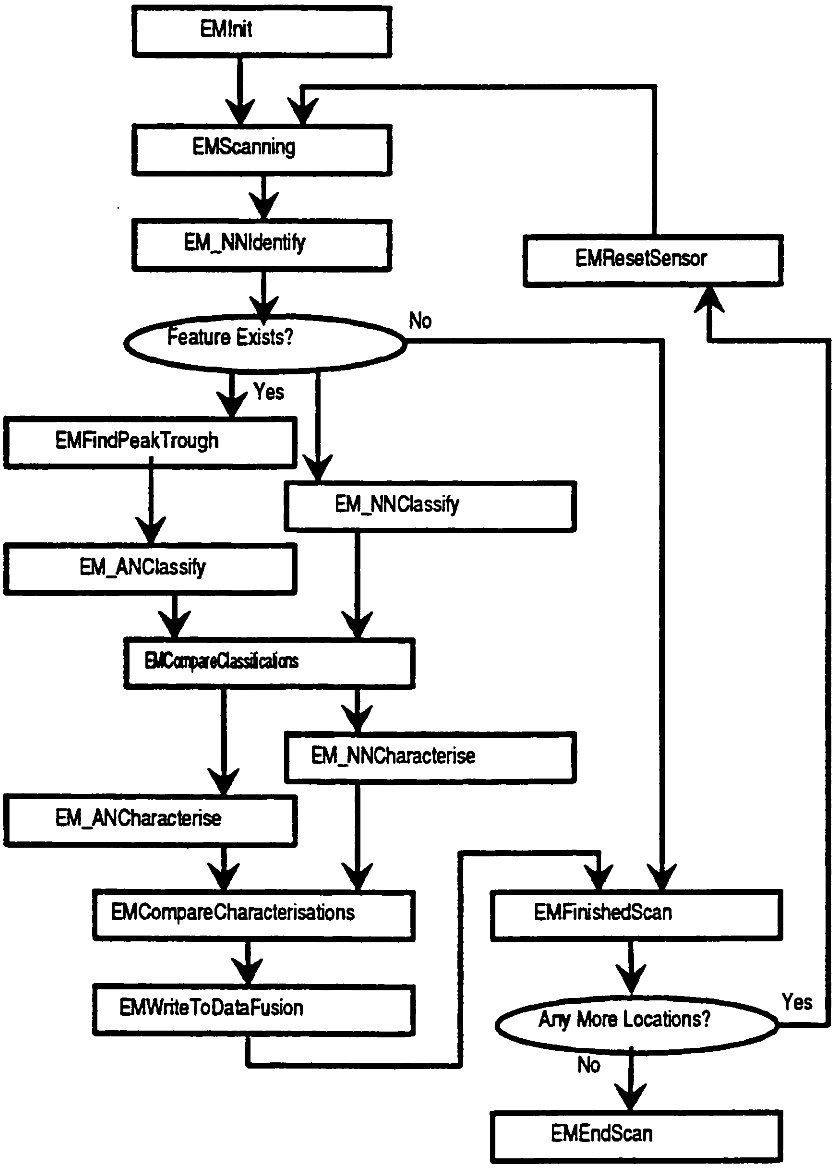


Figure 7.11 Flowchart of electromagnetic sensor KS operations

Any feature is simultaneously classified and characterised by EM_NNClassifyCharacterise, which is shown in two parts in the flowchart, and classified only by EM_ANClassify. The results are stored in the instance of EMFeature and EMCompareClassifications compares the classifications. The feature is then characterised using look-up tables based on ACFM theoretical modelling by EM_ANCharacterise. The characterisation is then compared with that of the earlier result from EM_NNClassifyCharacterise by EMCompareCharacterisations. The final results are now written to the common data fusion format, i.e. to an instance of DFFaultData by EMWriteToDataFusion. It also indicates that the process of investigating the feature has been completed.

If no feature is found, the KS EMFinishedScan is triggered, to determine if any more sensor scanning of the surface should be performed. The electromagnetic sensor KBS may then loop back on itself to re-scan the surface, or it may terminate its operation.

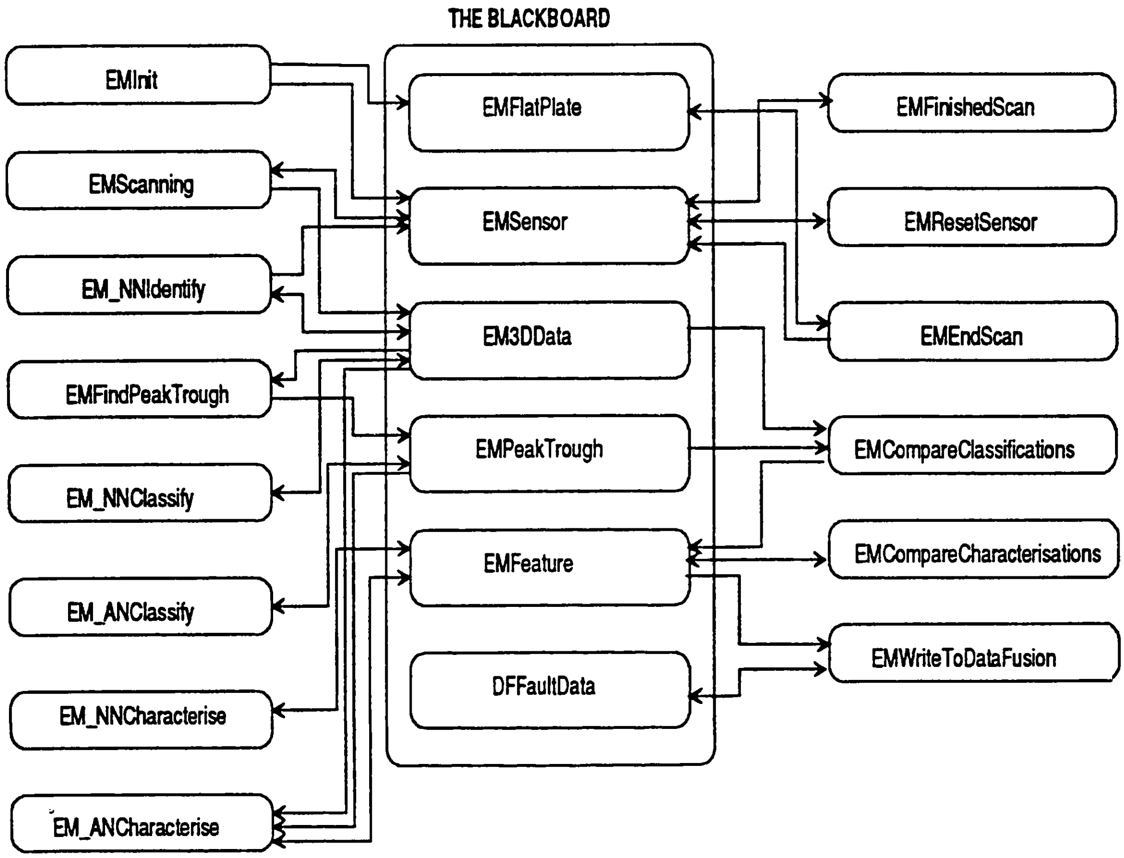


Figure 7.12 Interaction of electromagnetic sensor KSs and the blackboard

The following details the implemented KSs, but also describes the KSs required for a general system which may be considered in future development.

Some example KS Cards are given in Table 7.4.

Table 7.4 Example KS cards

knowledge source	**EMInit**
activation condition	
input	
output	state, geometry, now, parameters, location

action	CREATE x IN EMFlatPlate WITH x.state = state, AND x.FGEOMETRY = geometry; CREATE y IN EMSensor WITH y.state = ready AND y.inspects_specimen = x AND y.location = location AND y.time = now AND y.another_location = notyet AND y.SENSORPARAMETERS = parameters
knowledge source	**EMANCharacterise**
activation condition	EXIST x IN EMFeature WITH x.an_characterised = no EXIST t IN EM3dData AND u IN EMPeakTrough AND v IN EMSensor WITH x.has_data1= t AND t.resolution= high AND u.has_data= t AND t.with_sensor = v
input	u.aPktrData, v.SENSORPARAMETERS, x.type
output	pit_parameters, crack_parameters
action	MODIFY x WITH x.an_pp = pit_parameters AND x.an_cp= crack_parameters AND x.an_characterised = yes

7.3.3.1 EM Sensor Control KSs

Since EMInit is the first KS to be activated, it has no activation condition nor input parameters for the treatment function. It initialises the blackboard by creating instances of the component, and of the electromagnetic sensor with attributes taken from a file.

The EMScanning KS collects pre-processed data from the sensor by reading, from a particular point in the component description file, the **B** data. Two matrices of electromagnetic sensor data are collected for two different resolutions.

The EMFinishedScan KS inspects the list of points to be scanned in the electromagnetic sensor inspection plan to see if there are any more points left to scan. If there is another location to be scanned, the EMResetSensor KS creates a new instance of the electromagnetic sensor ready to input new data. If no more locations need to be scanned, the electromagnetic sensor KBS ends operation with EMEndScan. For this KS, no associated C code was required.

EMError can interpret any error codes for the electromagnetic sensor KSs. At this stage it merely reads the error code from the blackboard and any text associated with is output to the MMI.

7.3.3.2 Electromagnetic Sensor Signal Interpretation KSs

There are two main operations in the KBS: inspection of specimen geometry and inspection of specimen features. The geometry of the specimen must be known to perform adequate inspection of the specimen. The geometry is supplied as part of a component description file which also contains the stored sensor data.

The EMFindPeakTroughs KS uses the chosen search algorithm to pick from the **B** matrices the two greatest $\mathbf{B}_x$ values ($B_{xmax}1$, $B_{xmax}2$), the $\mathbf{B}_x$ minimum (B_{xmin}), the two greatest $\mathbf{B}_y$ values ($B_{ymax}1$,$B_{ymax}2$), the two least $\mathbf{B}_y$ values ($B_{ymin}1$, $B_{ymin}2$), the $\mathbf{B}_z$ maximum (B_{zmax}), and the $\mathbf{B}_z$ minimum (B_{zmin}). A KS for identifying possible features, EMInspect, was created as part of the Phase I AIRES system. It analysed the magnetic field matrices $\mathbf{B}_x$, $\mathbf{B}_y$, and $\mathbf{B}_z$ for the possible existence of features, by testing that they satisfied the following

1. $B_{zmax} > 0$, $B_{zmin} < 0$
2. The value of $\mathbf{B}_z$ at the nearest point in the sensor array to the halfway point between B_{zmax} and B_{zmin} should be approximately 0, that is less than 10% of B_{zmax}
3. $B_{xmin} < 1$
4. $B_{xmax}1 = B_{xmax}2$, to within an appropriate tolerance

Unfortunately, tests carried out showed that the analysis employed could not easily distinguish features. Instead, a neural net was developed which could carry this task out very effectively. Thus this KS was not part of the final implementation and was replaced by a KS with a neural network.

If an irregularity exists it must be classified; it will be one of an agreed list of flaw types, that is pit, crack. or unknown. The EM_ANClassify KS determines whether a feature is a pit or a crack if EM_NNIdentify indicates that a feature exists. The algorithm to determine the feature type is based on the positions of the four $\mathbf{B}_y$ peaks and troughs. As illustrated in Figure 7.6 on page 270, these are placed around the feature: the acute angle α between the two lines joining these points is measured. For a pit, α is close to a right angle; for a crack, it is close to zero. Currently if the angle is less than 30°, the feature is classified as a crack; for angles between 65° and 115°, a pit is recorded.

Once the feature type has been determined its details, such as its depth and length, are extracted by EM_ANCharacterise. The grading of the feature is done after fusion of the results of the analytical methods and neural nets.

7.3.3.3 Data Fusion KSs

In the final implementation of the AIRES system, data fusion may take place at two points:

1. results from the EM neural nets are combined, if possible, with the results from the KSs based on analytical theory

2. the final output from the electromagnetic sensor and vision KBSs are compared

In the first step, only the one KS, EMWriteToDataFusion, is required to ensure that the results can be compared between KBSs. No grading of classifications is carried out as yet.

Classifications carried out by the neural net KS, which uses EM3dData for this, could be compared to the analytical method KS, which would use the higher level data in EMPeakTrough. In Phase II, before it was realised that the use of the analytical method would lead to problems, it was proposed that a simple comparison would be carried out by EMCompareClassifications. If there is any disagreement at this stage then the result would be an unknown classification. This of course was discarded as an option at this stage for real data, but was possible with artificially generated **B** data. In future work, it may still be possible to use higher level reasoning.

After characterisations have taken place, EMCompareCharacterisations combines the results as simple averages. The EMWriteToDataFusion KS transfers the information held on features identified by the EM KBS to the instance of DFFault, in order that the data fusion process can be completed. The data fusion process was very basic: it involved presenting the user with the output from each independent KBS. The intention was that at a future date the final result from each KBS would be combined using some Bayesian decision process based on *a priori* probabilities, but the idea is fundamentally weak in that no use is made of intermediate results. This is further discussed in Section 7.3.5.

7.3.3.4 User interface KSs

The user interface's main function is to inform the user of the operations of the sensor and KBS throughout. The main KBS is expected to operate such system features as user and geometry input as well as the grading and output of results. As the specimen is examined by the electromagnetic sensor the KBS will build up a picture of the specimen geometry. The user should be shown what the KBS "sees".

7.3.3.5 Future KSs

Other KSs which were not implemented, but which may be required to allow greater control of the sensor and manipulation of the sensor data.

A scan may be performed on a specified surface area of the specimen, either the whole surface, or some part of it. The data sought by the scan consists of the *x, y* or *z* components of the magnetic field at the surface of the specimen. In the current implementation, a scanning route, has been predefined. In future implementations it can be foreseen that the KBS will be able to re-direct the sensors to either carry scanning as required or even to re-scan problem area by EMRequestSensorScan.

To allow complete flexibility in the method of collecting data, a KS EMDataCollect, which carries this out explicitly is required. Data may be collected from the electromagnetic sensor using a serial or parallel communication interface or from previously stored data. The data will be filtered through a pre-processing routine before being fed to the electromagnetic sensor data interpretation functions.

The KBS must be able to complete the feedback loop to control the sensors by altering the various sensor parameters by use of EMSensorAdjust. The parameters in the electromagnetic sensor operation that may be adjusted are the gain, the frequency, the solenoidal current and the sensor lift off. In the current implementation, pre-processing of data is carried out before the data reaches the KBS. It can be foreseen that in the future, the KBS may require more direct control over pre-processing parameters and thus would require an EMPreprocessData KS.

Before analysis of the **B** matrices supplied by the electromagnetic sensor control functions, the data must be pre-processed to account for noise, variations in the coils used to measure the magnetic field, and the non-uniformity of the inducing electrical field. Pre-processing is initialised by a calibration process KS EMCalibrate which is performed on a smooth featureless sample of the material to be inspected. This initial calibration need only take place once for each material type.

In the future the geometry of the specimen may be supplied to the electromagnetic sensor KBS by the main KBS, which in turn will receive it from user input or vision KBS or through pre-scanning. The electromagnetic sensor KBS may be required to check the geometry. This is a simple scanning and analysis operation compared to the inspections of specimen features and provides assurance that the electromagnetic sensor KBS and vision sensors are operating successfully. An EMCheckSpecimenGeometry KS would check the alignment of the sensor and main KBS, to ensure sensor operation is reliable. This would analyse EM data to determine specimen geometry and compares this with input geometry. Once the geometry of a component has been satisfactorily established, the KBS may initiate a general search for features using EMSearchForFeature. This involves building up a quick picture of the overall component and involves combining several **B** matrices.

Other KSs may be implemented for direct output from the electromagnetic sensor KBS, that is, before data fusion with the vision KBS results.

- EMDepictSpecimenGeometry - The purpose of this KS is to inform the user of the KBS knowledge of the specimen's geometry by calling a graphics routine to draw a model of the specimen.
- EMIndicateSensorOrKBSOperation - The KBS must tell the user what it is doing; this will include giving statistics of the process being undertaken and appropriate visual or other data.
- EMDepictFeatures - When a feature is being investigated the KBS must inform the user of the feature's characteristics. Once the existence of a feature has been firmly established, it will be shown in the specimen geometry.

7.3.4 The Use of Neural Networks

The original intention was to carry out all electromagnetic sensor data interpretation explicitly, that is, by use of knowledge sources encapsulating explicit reasoning. This was based on the experience of human experts who were able to recognize the existence of a feature in the electromagnetic data when viewing output from the ACFM sensor. As already mentioned, the knowledge sources based on the encapsulated and high level knowledge of surface shapes were not successful when considering real data.

The possibility of using neural nets for rapid interpretation for data was originally considered for the vision data only. The vision system produces measurements on the position, area, perimeter, and the second order position invariant moments of inertia of the possible defects detected. A neural net can be used to provide an initial classification of the defect into pit, crack, multiple defect, or unknown. Where the defect cannot be classified with confidence, additional knowledge sources may select image processing operations to provide more defect parameters.

Parallel work by co-workers in Deutsche Aerospace was then carried out on the use of neural networks for this problem. It was found that neural nets were very effective and more efficient than making use of the electromagnetic sensor theory for defect identification. Neural nets were then incorporated into the electromagnetic sensor KBS by the TRAM developers to provide a very effective method of checking the magnetic field data for a possible feature as EM_NNIdentify. This became the only KS in the EM KBS which identified features. Additionally, neural nets were developed which could classify and characterise a feature from low resolution data simultaneously, and these were included in EM_NNClassifyCharacterise. This KS carries out classification and characterisation in parallel to the analytical knowledge sources.

This result corresponds well with experiences in the AI field where human experts cannot always explain how they recognize a situation. It is clear that this is a pattern recognition problem which cannot be solved easily by symbolic high level reasoning. Future work would concentrate on applying neural networks to this problem.

7.3.5 Issues of Data Fusion

The complete AIRES system is intended to use the data from the vision and electromagnetic sensors to provide a full defect inspection of a particular component. This involves data fusion, whereby data from different sources are somehow combined. In general there are two basic types of data fusion: early fusion where the merging of the data is done near the sensor, and late fusion, where the merging is performed later during computational analysis and processing of the data. The former situation is quite common where identical sensors are used; the latter is more usual if the data comes from different sensors.

In AIRES, since most of the work was concentrated in automating electromagnetic signal interpretation, the main form is that of late data fusion. The final results of the electromagnetic sensor KBS and of the vision KBS are combined. When, for example, the electromagnetic sensor KBS has located, classified, and characterised a defect, this result must be checked with the vision KBS. The flowchart of Figure 7.13 shows how this is accomplished.

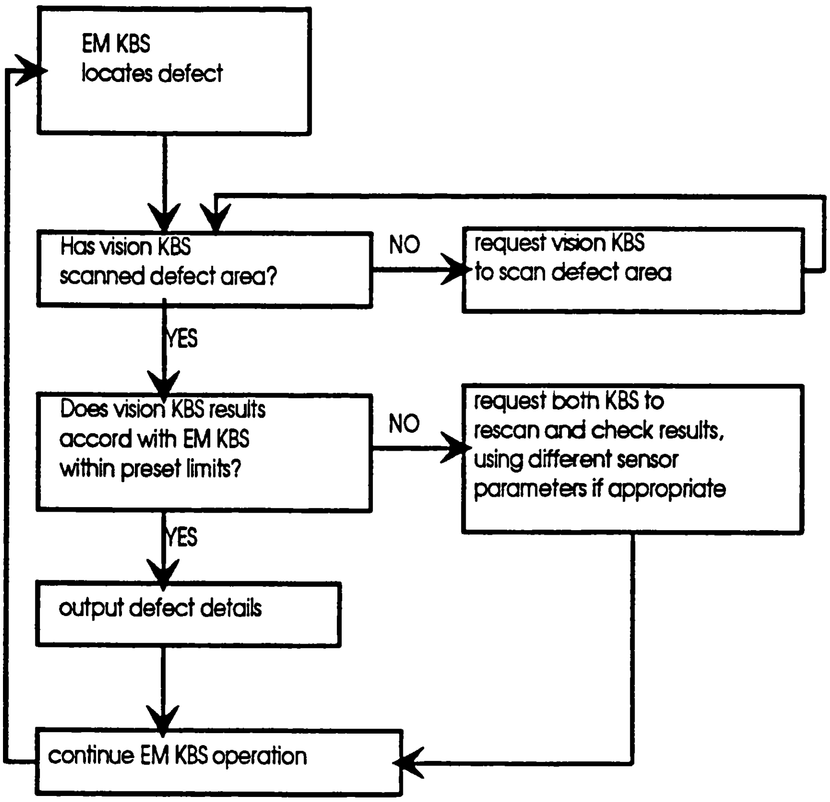


Figure 7.13 Illustrating AIRES data fusion

In addition, some early data fusion takes place between the neural net KSs and the KSs based on analytical methods for the electromagnetic sensor KBS. Thus the implemented data fusion process is rather simplistic. In the following, issues relating to data fusion are discussed, in order to consider how the work carried out may be extended to provide more intelligent data fusion.

7.3.5.1 Asynchronous Use of Sensors

In the current system, the two sensors do not necessarily scan the component in the same order, and defects may not be detected simultaneously. Thus the data structures or objects stored in the blackboard need to incorporate information on the time of detection and location of possible defects.

Suppose a particular surface area has been inspected by each sensor. If the two subsystems differ in their interpretation, further information must be obtained to resolve the KBS inconsistency. This may be achieved by changing the operational parameters of each sensor to obtain different data from the same area. In the event of an intractable inconsistency, the operator may be informed who then takes the final decision.

A further possibility that must be considered is that one system may fail to interpret the data received. In this case, it may request that the other sensor scans the originating area to provide more data to assist its interpretation. As an example of this, consider that the vision subsystem detects what appears to be a pit. The vision subsystem knows that it will not be able to measure the depth of the pit and thus may request other sensors to inspect the pit in order to find the depth. The request can be made even if the vision sensor has no knowledge of another sensor existing and may can take the form of a "mayday", that is, a call to any sensor listening to inspect the area where it has not been able to identify the feature. Data fusion becomes an iterative process with the aim being to obtain convergent KBS results.

7.3.5.2 Combined and Irregular Flaws

Another question which has not yet been raised is that of combined or irregular flaws. One example is that of a defect which does not lie at the centre of the scanning area. Noisy electromagnetic sensor data may not allow for detection of such a feature. If another sensor detects such a feature, then as part of data fusion, re-scanning may occur to cover the area containing the feature. It should be noted that there will be a considerable difference between the inspection of a nearly featureless plate with, say, a single crack and a single pit well separated from each other, and of a surface which is massively corroded and scratched with numerous flaws of all types. In the former case, the magnetic field will be well structured and easily receptive to analysis; in the latter, isolation of features at the pre-processing stage alone will be a considerable task. Even the difference between the investigation of a single semi elliptical crack and a single crack with an irregular profile, which may consist of a combination of semi-ellipses, will be significant. The main case to consider is where features combine, for example a crack overlaid by a pit. Even if such a flaw is surrounded by flat and flawless surface, the analysis of the magnetic field of the combined flaw presents a further problem.

Another simpler example can be given: supposing that the vision sub-system has identified some surface defect, which may be a pit. If the electromagnetic sensor KBS cannot detect nor characterise this defect, then this would indicate that the surface defect may be merely paint on the component rather than a defect of the metal.

For a general system, the data fusion design should be extended to allow for re-scanning, split features, and inter-KBS communication for combined features at scanning time.

7.3.5.3 Opportunistic Partial Interpretations

The data fusion process may also allow opportunistic inspections. This is done by ensuring that intermediate results about possible detected features and classifications are stored in the blackboard. These may also be combined with the knowledge about the component to allow targeting of areas to inspect.

In the last case given above, instead of the vision subsystem making explicit requests to another sensor to take over, the vision subsystem could post all its intermediate results onto the blackboard and if it cannot continue interpreting data for a region of the component, it will then move onto to consider another region. Any other sensor, an ACFM sensor, say, would recognise that the blackboard contains the information that a pit of unknown depth has been found and would proceed to inspect the area to size the pit without passing through the detection and classification processes. Further, if several defects have been detected in many areas of the component, then the knowledge sources required to characterise those which are in the more critical areas, leading to complete rejection say, as opposed to re-working of the component, would be triggered first. Also, if necessary, the sensors could be requested to re-scan those areas, before inspecting other areas.

By combining two or more sensors and providing data fusion at all levels, it is possible to enhance the capabilities of the inspection techniques, both in terms of the speed at which defects are detected and characterised and the types of defects that can be identified accurately. In addition, the use of a blackboard architecture may also allow future work in this area to consider providing additional knowledge bases on the component being inspected and the likelihood and relative importance of defects at different points of the component. This form of knowledge which could enable automatic inspection systems to be more powerful and useful.

7.3.5.4 Data Fusion Concepts

It can be argued that the implementation of early data fusion may lead to a conflict of interests. It was stated earlier that one of the benefits of the blackboard concept is that it allows separate teams of experts to work independently in order to provide sets of knowledge sources representing their areas of expertise. Early data fusion may require these teams to combine their knowledge to provide knowledge sources for the data fusion process, thus the whole procedure no longer allows for complete independence. This argument can be overcome by noting that a set of common blackboard concepts that trigger data fusion KSs will be useful. Additional concepts may be included to allow data fusion at the early stages of identification of defects.

7.3.5.5 Expert Knowledge as part of Data Fusion

It is in places such as this that the appropriate inclusion of expert knowledge is vital, and where in fact this system transforms from a merely analytical system to a KBS. At this step, most of the data fusion process is carried out so that the results from the visual sensor can be compared with that those from the electromagnetic sensor to give information indicating if there are possibly combined or irregular features.

7.3.6 Implementation and Testing

The knowledge base systems in AIRES were implemented using blackboard-based software, TRAM, that is an AI tool originally developed using LISP originally for robotics applications. It was then translated to ANSI C on Sun workstations. For AIRES, work was carried out to port TRAM to IBM RS/6000 workstations with the AIX operating system by and the electromagnetic sensor KBS Knowledge Source functions were written in ANSI C. The implementation procedure was as outlined in Figure 7.14.

The diagram indicates the allocation of tasks for the electromagnetic KBS to each partner involved in the AIRES project: LETI developed TRAM and encoded the agenda and concepts; DASA developed the neural networks which were incorporated by the University College London team into appropriate KSs; UCL provided specifications for the concepts and agenda, coded the C functions for the KSs and ported the TRAM C code to the IBM RS/6000 workstations.

The electromagnetic data transfer from the ACFM sensor to the AIRES software was carried out initially through files only. The two datafiles were component description file containing the specimen geometry, and a sensor data file holding the magnetic field data as in Figure 7.15. The process could be easily modified to allow real-time data transfer via an RS232 link.

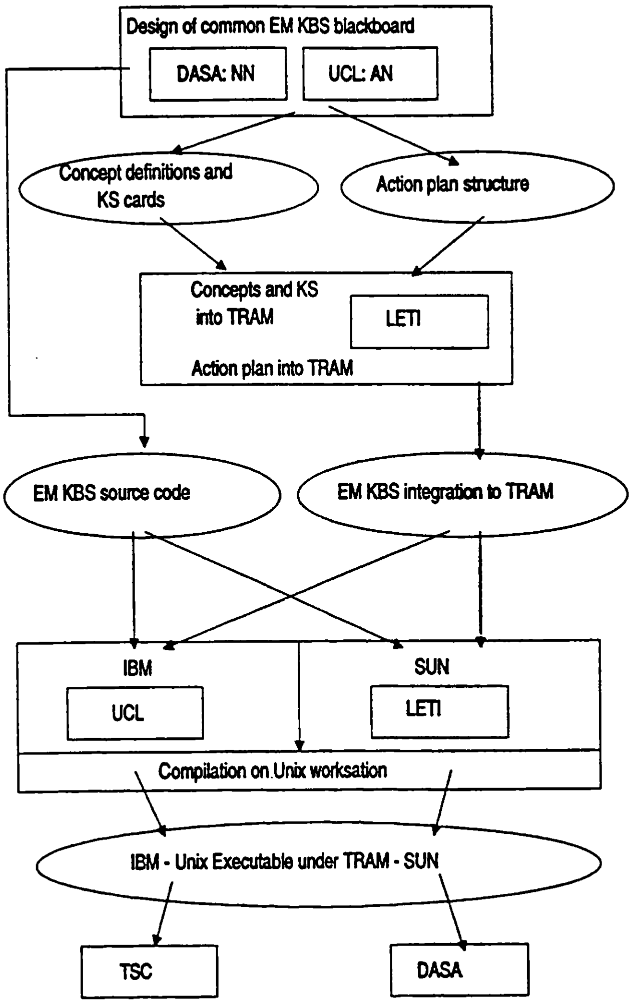


Figure 7.14 Electromagnetic sensor KBS implementation process

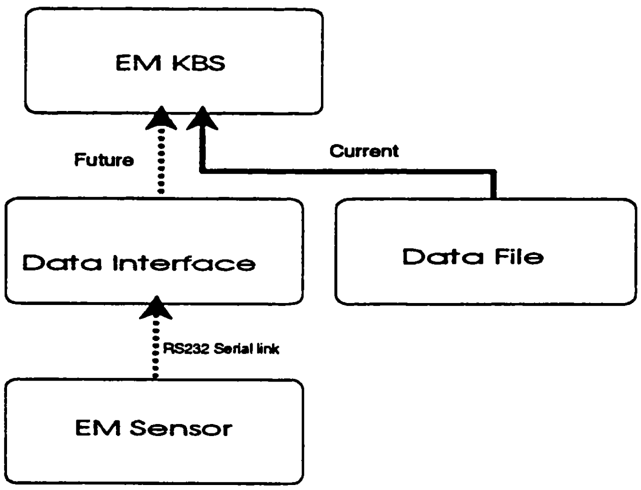


Figure 7.15 File-based data transfer

7.3.7 Tests for Phase I Detection and Categorisation

Tests carried out on the KS treatment functions were carried out using artificially generated data and with data recorded from sensor scans. The features analysed in these tests were pits and cracks on flat plates only.

7.3.7.1 Theoretically Generated Data

The analytical knowledge sources of the electromagnetic sensor KBS depend for their operation principally upon the accurate location of the turning points of the **B** data. The artificially generated data is noise-free and has clearly defined peaks and troughs. Consequently the electromagnetic sensor KBS functions with reasonable accuracy.

The crack lengths found from the data were within 10% of the actual crack lengths and the pit radii are also reasonably accurate. The crack depths were not nearly so accurate. This can be attributed to the fact that although the error in the position of the actual peak or trough co-ordinates would be only one half-unit either way of the computed co-ordinates, the values of the turning points are likely to be significantly different from the values taken read off the **B** data owing to the sharp gradient of the **B** surface near these points. This problem may be overcome if the peak/trough location algorithm is changed to surface-fitting procedure as this would probably provide a better estimate of the value of the function at the estimate peak/trough point.

7.3.7.2 Real Sensor Data Results

The real sensor data is much more distorted than the artificially generated data and the current electromagnetic sensor KBS methods were not able to analyse it successfully. In particular, the peak and trough searching algorithm was applied only to masked data, since it did not work at all for unmasked data. This algorithm did not always find the correct number of peaks and troughs.

Some problems which affected the implementation of KSs for the electromagnetic sensor KBS were related to the physical limits of the ACFM probe. The sensor coils used had a diameter of 1mm which limits their resolution. For example, 10mm pit defects should have a clear effect on the recorded magnetic field, 5mm pits a less clear but noticeable effect, and 2mm pits may be hardly noticeable, especially after the smoothing from masking. Additionally, the theoretical work assumed a uniform, that is, constant inducing field, but the actual inducing field is non-uniform.

As electromagnetic induction is a non-linear effect, the variation in the field cannot be removed by simple linear correction. Even cases where there are clearly defined peaks and troughs, the field non-uniformity makes it difficult to determine the desired maxima and minima. The locations of the peaks and troughs may also be displaced. Hence either a uniform inducing field must be provided, or some other way of making use of explicit knowledge, say in the form of a set of rules, must be devised to identify the turning points of interest.

The first solution may be practically impossible to ensure in a usable sensor, although some work was carried out with the AIRES project by colleagues to develop an improved probe. Yet the latter solution may not produce results as good as those produced by the analysis of uniformly induced data.

7.4 CONCLUDING REMARKS

The finished AIRES system works for the given component and with several simplifying assumptions being made. Further work is required to extend the system to one which can be used in real applications.

For a general component, a general 3D co-ordinate system should be implemented, together with identification of component geometry. The former will then allow more complex geometries to be considered; the latter will enable the system to be applied to perhaps several components at any one time, or more importantly be used to ensure that the component has been placed properly in the required position. The concepts used here should be re-designed to follow a format based on the STEP description language for product data storage, or other similar data storage standards to allow future extensions of this work to other industries in manufacturing.

For a more effective use of multi-sensors the following improvements should be considered:

- more extensive data fusion at many levels
- real-time control of sensors, whereby the results from one may affect the use, such as where to place the sensor or the settings of operational parameters, of the other sensor

These points will enable more rapid and efficient classification and sizing of possible features. For improved use of the electromagnetic sensor, in particular, the following issues require further work:

- addition of a rule-base to assist turning point location or of advanced pattern recognition techniques to aid feature detection
- assembly of several **B** matrices to allow identification of features larger than the probe head size or of partially scanned defects

One of the major weaknesses of the tool-kit of electromagnetic sensor knowledge sources now developed, is that the KSs are sensitive to the chaotic nature of real data. With improved sensor design and the addition of rule bases to enable expert interpretation of the data it is expected that this problem may be overcome.

The AIRES concept involves linking multi-sensor (ACFM and vision) data with fast modelling to give real-time signal inversion and fusion of data within a black-board system to interpret the reconstructed images. The AIRES prototype system provides a basis for a general automated inspection system.

The AIRES concept is important to ensure that automatic interpretation of sensor data based on the component geometry, expected defect types and other background information. AIRES-like systems could be combined with analysis and decision support systems, such as the RISC System, which require consistent interpretations of inspection data.

8 CONCLUSIONS

The purpose of this work was to demonstrate the feasibility of providing operators of offshore structures a computer decision-support tool for inspection scheduling. The objectives were to develop a suitable methodology for scheduling based on the output from structural reliability analysis combined with fatigue fracture mechanics and to implement a demonstrator knowledge base system for Reliability based Inspection Scheduling (RISC).

The work carried out in developing the knowledge base system has shown that the use of advanced software techniques when combined with classical analytical programs, such as the FORTRAN program, RISCREL, can provide substantial benefits in ease-of-use and decision making when there is a large volume of information and many different types of data. Extensions to the RISC work for general inspection scheduling are discussed here and concepts for the integration of an AIRES-like system with the RISC System are described.

8.1 BACKGROUND TO THE WORK

In the offshore oil industry, the current statutory requirements in the UK are that the certifying authorities issue a Certificate of Fitness for each platform according to The Offshore Installations (Construction and Survey) Regulations (1974) (MTD, 1989). The operators of the structures are required to ensure the integrity of these structures by carrying out periodic inspections and repairing when necessary. Decisions on inspection, repair and maintenance (IRM) actions on offshore structures were based on engineering judgement and these decisions were supported by the use of deterministic analyses.

The work towards the RISC methodology started in 1990. At this time, there had been many advances in the areas of

- structural reliability analysis and the Level II methods FORM and SORM
- fast analytical models for fatigue assessment of tubular joints
- corrosion effects on fatigue crack growth
- reliability of inspection techniques

The use of structural reliability methods combined with the other advances on corrosion fatigue modelling and inspection reliability can lead to more rational strategies for scheduling of IRM actions. Applying structural reliability analysis to inspection schedules, however, requires understanding the inspection procedure and making use of the appropriate information on inspection techniques. There are difficulties in collecting input data. The interpreted results need to be combined to form a global solution for the structure that takes into account practical constraints. These issues point to the use of artificial intelligence (AI) techniques to provide operators with a decision support tool to aid scheduling. Many advances in the field of artificial intelligence have been made in

- knowledge representation to support reasoning processes
- general and systematic searching techniques
- methods of dealing with hard and soft constraints

Several AI -based systems have been developed for planning and scheduling for general human reasoning, manufacturing, and robotics and control. A few researchers have considered the maintenance planning problem and in general, such work has concentrated on production lines. To date, no system targeting offshore structures integrates objective analysis methods with a flexible, but systematic scheduling algorithm.

8.2 SUMMARY OF ACHIEVEMENTS

The main aim of this thesis was to pull together techniques from several disciplines to solve a complex, real-world problem, that of scheduling of maintenance actions for a complex structure. This work extended knowledge in applying pure AI techniques to help provide a framework for integrating structural reliability analysis, fatigue fracture mechanics, and planning and scheduling which takes into account guidelines and resource constraints.

- **Reliability based Inspection SCheduling for fixed offshore structures (RISC)**

The RISC methodology aimed to integrate structural reliability with fatigue fracture mechanics to enable rational scheduling of IRM actions. The RISC Demonstrator has the following features:

- the object base contains all the data required to carry out the ranking, the analysis and the scheduling of IRM actions for a fixed jacket platform
- rapid prioritising of joints is carried out based on the values of attributes associated with each joint
- the execution of COMREL, the early version of RISCREL, is controlled by the KBS
- the expected costs for each IRM action are used to produce an initial schedule

- searching techniques and heuristics are used to modify the initial schedule in order to take into account resource constraints, which may be in turn reallocated by the user to enable "what-if" analysis

The achievements of the author in this work and as described in the Chapter 5 and demonstrated in Chapter 6 are:

1. A through review of the IRM procedures and requirements was carried out and this provided basic information for the specifications for the reliability based fatigue fracture mechanics and the decision theoretic basis for the analysis.

2. A prototype knowledge base system was produced which can schedule based on reliability index and expected costs only and this was extended to build the RISC System demonstrator.

3. Documents of the detailed design for the interfaces for RISCREL, the reliability based fatigue fracture mechanics software, and user interface specification and design were produced (Dharmavasan et al, 1994b).

4. Knowledge base system modules for system control, ranking joints to prioritise joints for analysis, planning possible maintenance plans for a joint, and constraints-based scheduling of the IRM actions have been developed.

5. A case study has been produced to demonstrate the use of the RISCREL software and the constraints-based scheduling algorithms.

- **Automated Image Recognition using Expert Systems (AIRES)**

The electromagnetic sensor KBS of the demonstrator AIRES system includes a set of knowledge sources which can be used as a toolkit for basic electromagnetic sensor data interpretation. The implemented prototype has the following features:

- it can be used to inspect flat and cylindrical components
- the component geometry is provided in the component description file
- the sensor data is pre-scanned and stored in the electromagnetic sensor data file
- basic data fusion is also allowed

The achievements by the author are:

1. Appropriate concepts and the basic procedure for defining knowledge sources based on the inspection procedure will act as useful starting point to future work.

2. The electromagnetic sensor KBS subsystem represents a step towards understanding how

electromagnetic sensor signal interpretation may be carried out automatically and intelligently for multi-sensor inspection tasks.

8.3 FUTURE WORK

In order to apply the RISC System in practice, the current demonstrator system requires some immediate work to allow operators to use it directly:

1. Operators would require the ability to be able to interface the RISC System directly with other existing information systems, such as inspection databases, to provide the RISC System with all the necessary information. Interfaces to operators' own information systems are needed.

2. The ProKappa code should be translated to a language which can be ported to different hardware. This work would be essential before possible commercialisation.

3. Currently only text files are used as interfaces to the analysis modules and for set-up and these are predefined in format. Different file types and formats should be allowed in the RISC System. Modules for interfacing to external analysis modules should be developed to allow set-up data from any source. In particular, interfaces to finite element systems are required. A generalised language for set-up should be considered which allows file formats to be defined by the operator or by automatic definition to interface to external analysis modules.

4. A detailed design for a MOTIF Windows-based graphical user interface was produced during the RISC project. This should be developed fully and implemented for the RISC System.

5. The integration of AIRES into RISC requires at least an interface to allow the final interpreted results from an AIRES-like inspection system to be added to the RISC object base.

8.3.1 Analysis of the Structure

The structural reliability analysis in RISC has been restricted to Level II with a limit state based on fatigue crack growth only in tubular joints. This presupposes many assumptions which may not be valid for all components. In addition, the structural reliability analysis may be extended in many ways.

1. One fundamental problem is that it is assumed that only one crack is of interest in the tubular joint under consideration. There is the problem of groups of cracks and how to compare detections and measurements against the actual cracks. Consider the real cracks shown in Figure 8.1(a). An inspection may reveal cracks as shown in 8.1(b) or as in (c). Neither of these results represent a poor inspection result since they are indications the actual cracks.

a)

b)

c)

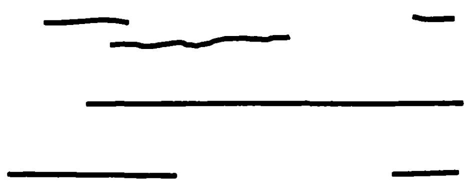


Figure 8.1 Crack classification

There are several possible ways of defining what constitutes a crack and its size. The approach taken by inspection reliability experts is to classify overlapped cracks and cracks a very short distance apart to be one. The justification for this is that cracks which are close together do behave as one, due to the high stress concentration between the two cracks leading to very rapid crack growth. It has also been assumed that only one crack per weld would be considered. Further studies could be carried out to see how different classifications of cracks and multiple cracks will lead to different reliability measures and hence the scheduling of inspections.

2. Some of the base assumptions for the development of the decision procedure implemented as part of RISCREL need to be investigated carefully. For instance, the reliability assessment is based on the assumption that repair is carried out immediately if a crack is found. As already discussed, a possible outcome of the use of the RISC methodology is that repair may be delayed if it can be shown that this will not affect the *fitness for purpose* of the joint or structure. Not so serious from the point of view of reliability, but of practical significance, is that the cost of repair will vary considerably according to whether or not repair is carried out at the same time as inspection, or if several confirmatory re-grindings are required. Yet the costs are treated as constants. Sensitivity analyses should be carried out to gauge the extent to which the process is affected by variations and uncertainties in the costs. Such an investigation would require access to potentially sensitive financial information.

3. Consideration of failure modes other than fatigue will require the development of several RISCREL-type programs to carry out structural reliability analysis based on different failure functions, modelling each required failure process. The possibility of allowing other damage mechanisms to be considered is discussed below in Section 8.3.4. In the short term,

heuristics would be required to deal with damage mechanisms for which suitable models that can be integrated into FORM or SORM reliability analysis do not yet exist.

4. The RISC System would require a decision module to select the appropriate reliability analysis knowledge sources for each component. This system could act as a decision-aid tool for engineers which would incorporate the many analysis programs used by engineers together with the associated knowledge to be able to make use of these programs effectively. It would act as an intelligent information retrieval system, that is, as a repository for undocumented data, techniques and procedures.

5. Reliability measures for a structure may be combined to obtain immediate information on the criticality of the component to the structure. Systems reliability methods, such as the branch-and-bound and β-unzipping methods outlined in Chapter 3, should be considered for incorporation into the RISC System. Alternatively, systems reliability software packages such as RASOS may be combined with the RISC System (Gierlinski et al, 1993).

8.3.2 Scheduling Algorithms

The scheduling procedure is based on simple brute force searches through a scheduling tree representing a single structure. This may be extended in many ways.

1. It may not be useful to carry out an exhaustive generation of schedules based on all conceivable searching algorithms. The reason is that the schedules are based on the cost evaluation result for each inspection task. Yet, the cost of inspection is of course affected by the number and types of inspection tasks carried out in any one year, so the schedule itself affects the costs assigned. As was noted in Chapter 6, the failure cost has a huge impact on the schedule, thus for each node, the failure costs must at some point be determined with an accuracy that is currently not routinely established. Thus any perceived improvement in the quality of the schedule may not justify the extra work involved and may even give erroneous results. At the moment this is very much a grey area. An investigation should be carried out to find out how sensitive the total cost of overall schedule is to aspects such as the number of neighbouring joints included in one weather window, the use of many different inspection techniques in over one weather window, and so on. As before, this requires access to potentially sensitive information. In the meantime, systematic searches are effective tools to produce alternative usable and rational schedules.

2. Case-based reasoning is one artificial intelligence technique which has not been implemented for the RISC System, but which has been suggested for scheduling applications (Miyashita, 1995). If a proposed schedule is considered successful by the user, then an interactive and

inductive procedure may be used to obtain from the user some indication of how and why it is considered to be a "good" schedule. The information obtained can be used to update heuristics defining a "good" schedule or to make searches more efficient. Additionally, stored schedules may be used as initial schedules, rather than the cost optimal schedules from an initial interpretation of the cost evaluation output, to be modified according to other requirements or to take the reliability and cost evaluation data into account.

3. Offshore operators are often concerned with planning the maintenance across a field of oil platforms. The scheduling algorithms may be extended to considered many constraints across scheduling trees for several structures. This may involve setting up a new search tree for the field made up of scheduling trees for each structure and a new root node representing overall resource constraints for the operator organisation. The concern here is that, as this new tree would be very much larger, systematic simple searching techniques may no longer be sufficient as there may be a combinatorial explosion in the number of alternatives to search through. More complex constraint satisfaction algorithms would be required.

4. The current system follows a prescribed procedure. Any exceptions have to be identified as such and then must be treated separately. The use of the blackboard concept would allow the use of cooperating knowledge sources to carry out the most suitable analysis for each component and to reason using many criteria at different points of the decision-making process.

5. Data fusion combined with pattern matching could be used to reason about joints that are in some way similar but seem to be behaving differently within the structure, or to reduce the amount of individual analyses to be carried out. Comparing the service history of a joint with new analysis results for similar joints helps decide whether to inspect or not. The service history and latest analysis results combined with further information on the geometry of the joint and compared with other similar inspection decisions may help to select an inspection technique. Latest inspection results for the rest of the structure and past information on the joint, with information on the geometry of the joint will help decide what detailed analysis, if any, is required for the joint and will aid decision-making on any repair or future IRM action. Finally, learning from past cases of joints from other structures which have failed to try to identify factors indicating failure would cut down on detailed analysis and provide a greater understanding of causes of failure for the future (Stone et al, 1989).

8.3.3 An Integrated Monitoring, Inspection and Planning System

The RISC methodology is based on analysis which requires accurate models of inspection and loading. The loading information is more often assumed data used at the design stage. As already discussed, the inspection results are highly affected by the conditions under which the inspection is carried out. The AIRES concept may provide a way of ensuring consistent and accurate interpretation of inspection data. Additionally, offshore structures are subject to many different forms of damage mechanisms not just fatigue.

1. One improvement would be to have the most up-to-date environmental loading information to carry out the analysis with real data as opposed to design and simulated data. On-line monitoring is an obvious solution to collect data on sea-states, wave-heights, induced strains, levels of cathodic protection and crack sizes. This huge volume of data collected will need to be analysed, screened and interpreted into a form suitable for reliability analysis. Using on-line monitoring was first proposed as part of an intelligent structural integrity assessment system which is a natural extension of the RISC concept (Dharmavasan et al, 1988b).

2. The requirement to have consistent inspection results input to the reliability analysis leads to the idea of incorporating an AIRES-type system which carries out automatic and consistent interpretation of inspection data. To achieve this, the system would need to include require automatic image reconstruction from data from several NDT techniques, such as ACFM and close visual inspection, together with expert interpretation, to make unequivocal decisions.

3. Damage mechanisms may be categorised as either those leading to continual deterioration, such as crack growth due to fatigue, or wall-thickness corrosion, or those producing sudden damage, such as collisions, blow-outs, impact, punch-through, or even fracture due to extreme loads. Analytical and numerical models exist or are in development which can predict the deterioration of a component due to most damage mechanisms in offshore structures. Systems reliability techniques can be used to model the effect of the loss of component on the reliability of the structure. The RISC System could be extended to include several damage mechanisms.

4. Another area of great research interest is that of using neural networks for both interpreting data from sensors particularly for subsea applications, and for storing sensor data for stereotypical defect cases. As the base sensor data cases would require high amounts of memory and rapid methods of retrieval, the use of novel methods of storage, such as holographic techniques, would need to be investigated (Tao et al, 1995).

5. Any future work in the use of neural networks for ACFM data interpretation will consider different ways of providing intermediate results and a more formal basis for data fusion with other sensor results. Intelligent and early use of data fusion may enhance significantly the

total capabilities of an inspection system by combining intermediate results from different sensors, knowledge about the inspection techniques and the component being inspected. Concentrating on the blackboard system approach with multi-level data fusion, may lead to sophisticated and intelligent automatic inspection systems.

These extensions would go towards creating an integrated system for the complete life-cycle of an offshore structure. It will aid design that takes into account reliability and maintenance issues over the life of the structure, by allowing extensive comparison with other structures. Once the structure is in operation, it will allow monitored data on the state of the structure to be compared to design data and the results to be fed directly into the maintenance planning and scheduling.

8.3.4 Other Applications

The RISC methodology may be applied to any structure made up of many similar components subject to fatigue failures. In the case of extensions to RISC proposed above, which allow several models for varying damage mechanisms and different components, there are few restrictions on the type of structure.

The AIRES concept is applicable to industries ranging from aero-engine/aerospace to nuclear, chemical processing and the offshore industry. Wherever inspection is an important aspect of the maintenance procedure, but the inspection data is difficult to interpret in a consistent and accurate way, then an automated inspection interpretation system is required.

Several structures have been suggested for the application of RISC and AIRES, such as air-frames and steel bridges. The following are of particular interest to the author:

1. Another type of offshore structure which suffers from fatigue is the jack-up platform. Jack-up rigs were intended for exploration drilling only, but they are often used during the production phase. This leads to operational problems, in particular the new requirement to carry out inspections *in situ*. As these structures are subjected to the dynamic loading, they are affected by fatigue cracks. Other failure modes, however, such as blowout and collisions, but primarily punchthrough, are of greater importance. thus several failure mode analysis modules are required for jack-ups.

2. Pipelines are another example of aging structures which require periodic inspections to assure their integrity. Some work to develop assessment packages for pipelines has already been carried out by other researchers, such as Barbian et al (1992), based on in-line inspections using pigs. Failure modes for pipelines which have been investigated include spanning of the pipeline over areas where the seabed is not flat, leading to bending and,

ultimately, buckling; metal loss due to corrosion; and fatigue cracks growing due to cyclic loading from pressure changes (Gresnigt & Van Foeken. 1990; Gresnigt et al, 1994). These analytical and inspection tools may be incorporated in a RISC-like system to provide IRM schedules.

3. Several other researchers, such as Kobbacy (1992) and Efstathiou (1996), have worked towards applying KBS for maintenance and scheduling of production and manufacturing systems. There are many examples of structures in manufacturing which require general scheduling of operations and of maintenance actions. Intelligent systems would enable improved and more effective running of the systems. The essential features of such systems for manufacturing are that they should allow objective reliability analysis, pattern recognition and model selection as in the RISC/AIRES combined system.

8.4 FINAL REMARKS

The work described in this thesis represents a rational approach to solving a real-world decision problem using artificial intelligence techniques with traditional engineering analytical methods. The aim was to integrate the rigorous and objective measures of reliability with practical requirements and subjective criteria for inspection planning for fixed offshore platforms. The main achievements were in

- the establishment of a rational methodology based on reliability based fatigue fracture mechanics analysis for fixed offshore structures
- the specification of a computer architecture which integrates the analysis programs with the information on a structure and knowledge about scheduling requirements
- the detailed design of computer data structures and interfaces for the storage of the data and information required for planning and scheduling of IRM actions
- the development of a scheduling algorithm based on simple brute force searching procedures and extended to include heuristics providing the value of the schedules and dealing with multiple constraints
- the overall specification and design of the knowledge sources and concepts for an automatic inspection system making use of an electromagnetic sensor

This work goes towards providing a general procedure for planning and scheduling maintenance of complex structures and systems.

REFERENCES

Addis, T.R., *Designing Knowledge-based Systems*, New Technology Series, Kogan Page, 1985

Ahmad, K., Langdon, A., Frieze, P.A., *An Expert System for Offshore Structure Inspection and Maintenance*, Computers and Structures, Vol 40, No 1, pp 143-159, 1991

Aiken, M.W., Liu, O.R., *Software Review- Nexpert Object*, Expert Systems, Vol 7, No 1, pp 54-57, February 1990

Alty, J.L., Coombs, M.J., *Expert Systems*, NCC Publications, Manchester, 1984

Anandhi, B., Vinze, A.S., Sen, A., *A Blackboard Architecture for Reactive Scheduling*, Proceedings of the 26th International Conference on System Science, Maui, Hawaii, USA, 1993

Anderl, R., Mendgen, R., *Modelling with Constraints: Theoretical Foundation and Application*, Computer-Aided Design, Vol 28, No 3, pp 155-168, 1996

Andersen, R.T., Neri, L., *Reliability Centred Maintenance - Management and Engineering Practices*, Elsevier Applied Science, 1990

Anderson, A., Kragh, E., *An Intelligent Input to Inspection Planning on Offshore Platforms taking Gross Human Errors into Account*, Proceedings of the 10th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 91, Stavanger, Norway, Vol II, ASME, 1991

Ang, A.H-S., Tang, W.H., *Probability Concepts in Engineering Planning and Design, Volume 1 - Basic Principles*, John Wiley & Sons, 1975

Ang, A.H-S., Tang, W.H., *Probability Concepts in Engineering Planning and Design, Volume 2 - Decision, Risk, and Reliability*, John Wiley & Sons, 1984

API, *Recommended Practice for Planning, Designing and Constructing Fixed Offshore Platforms - Load and Resistance Factor Design*, 1st ed., American Petroleum Institute, Washington DC, July 1, 1993

Apte, C.V, Weiss, S.M., *An Expert System Methodology for Control and Interpretation of Applications Software*, International Journal of Expert Systems, Vol 1, No 1, pp 17-37, 1987

Arockiasamy, M., Lee, S., *Computer-Aided Structural Engineering (CASE) Project: State of the Art on Expert Systems Applications in Design*, Construction and Maintenance of Structures, September 1989

Atabaksh, H., *A Survey of Constraint Based Scheduling Systems Using an Artificial Intelligence*, Artificial Intelligence in Engineering, Vol 6 No 2, 1991

Baker, M.J., Vrouwenvelder, A.C.W.M., *Reliability Methods for the Design and Operation of Offshore Structures*, Proceedings of the 11th Ocean, Marine and Arctic Engineering Symposium, OMAE 92, Calgary, Canada, 1992

Barbian, A.O., Beller, M., Wunderlich, K., *Integrity Assessment Using In-line Inspection Tools*, Oil and Gas in a Wider Europe, Proceedings of the 4th EC Symposium, Berlin 1992

Baron, N.S., *Computer Languages - a Guide for the Perplexed*, Pelican Books, 1986

Benjamin, J.R., Cornell, C.A., *Probability, Statistics and Decision Theory for Civil Engineers*, McGraw-Hill, 1970

Bentley, J.P, *An Introduction to Reliability and Quality Engineering*, Longman Scientific & Technical, 1996

Berens, A.P., Hovey, P.W., *Characterisation of NDE Reliability*, Review Progress in Quantitative Nondestructive Evaluation, Vol 1, Plenum Press, pp 579-585, 1982

Berliner, H., *The B* Tree Search Algorithm: A Best-First Proof Procedure*, Artificial Intelligence, Vol 12, No 1, pp23-40, 1979,

Berry, P.M., *SCHEDULING: A Problem of Decision-Making under Uncertainty*, Proceedings of the 10th European Conference on Artificial Intelligence, ECAI-92, Vienna, Austria, 1992(a)

Berry, P.M., *The PCP: A Predictive Model for Satisfying Conflicting Objectives in Scheduling Problems*, Artificial Intelligence in Engineering, Vol 7, pp 227-242, 1992(b)

Berry, P., *Uncertainty in Scheduling: Probability, Problem Reduction, Abstractions and the User*, Colloquium on Advanced Software Technologies for Scheduling, Digest No 1993/163, Institute of Electrical Engineers, London, 26 April 1993

Bertini, L., *Statistical Fatigue Properties for Structural Steels for Offshore Applications*, Proceedings of the 13th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 94, Houston Texas, US, ASME, 1994

Besse, P., Seridji, A., Pelletier, J.L., Pate-Cornell, E., Regan, P., *Advanced Risk Management System (ARMS) A Resource Constrained Normative Decision System for Offshore Platforms*, Oil and Gas in a Wider Europe, Proceedings of the 4th EC Symposium, Berlin 1992

Birkinshaw, M., Kam, J.C.P., Sharp, J.V., *The Use of a Risk based Approach to Assessing Structural Safety in Offshore Structures*, Seminar on Risk Base Assessment of Structural Systems, Technical Advisory Group on Structural Integrity, 1993

Björk, B.-C., *Intelligent Front-Ends and Product Models*, Artificial Intelligence in Engineering, Vol 6, No 1, 1991

Blockley, D., *Engineering Safety*, McGraw-Hill, 1992

Bloor, M.S., Owen, J., *CAD/CAM Product-Data Exchange: the Next STEP*, Computer-Aided Design, Vol 23, No 4, May 1991

Bobrow, D.G., Winograd, T., *An Overview of KRL, a Knowledge Representation Language*, Cognitive Science Vol 1 No 1, pp 3-46, 1978

Bolc, L., Cytowski, J., *Search Methods for Artificial Intelligence*. London: Academic Press, 1992

Bramer, M.A., Muirden, D., Pierce, J., Platts, J.C., Vipond, D.L., *FAUST - An Expert System for Diagnosing Faults in an Electricity Supply System*, Research and Development in Expert Systems V, ed. B. Kelly and A Rector, CUP, 1988

Brascamp, M., *Reliability Centred Maintenance: The Application of the MSG-3 Method in the Industrial Market*, Proceedings of the 10th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 91, Stavanger, Norway, Vol II, ASME, 1991

Bray, D.E., Stanley, R.K., *Non-Destructive Evaluation - A Tool in Design, Manufacturing & Service*, Revised Edition, McGraw-Hill, 1996

Brebbia, C.A., Walker, S., *Dynamic Analysis of Offshore Structures*, Newnes-Butterworths, 1979

Buchanan, B.G., Shortliffe, E.H., *Rule-based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, CA US, 1984

Buchanan, B.G., Feigenbaum, E., *Dendral and Meta-Dendral: Their Applications Dimension*, Artificial Intelligence, Vol 11, No 12, pp 5-24, 1978

Buchanan,B.G., *Expert Systems: Working Systems and the Research Literature*, Expert Systems, Vol 3 No 1, pp 32-50, January 1986

Chakrabarti, A., Bligh, T.P., Holden, T., *Towards a Decision-Support Framework for the Embodiment Phase of Mechanical Design*, Artificial Intelligence in Engineering, Vol 7, pp21-36, 1992

Chandra, N. (Navinchandra), Marks, D.H., *Intelligent Use of Constraints for Activity Scheduling*, Applications of AI in Engineering, Vol 1, Ed D. Sriram, R. Adey, Proceedings 1st International Conference, Southampton University, UK, Springer-Verlag, April 1986

Charniak, E., McDermott, D., *Introduction to Artificial Intelligence*. Addison-Wesley, 1985

Chatalic, P. Dubois, D., Prade, H., *An Approach to Approximate Reasoning Based on the Dempster Rule of Combination*, International Journal of Expert Systems, Vol 1 No 1, pp 67-85, 1987

Chen, K., Zhang, S., Huang, W., *Artificial Intelligence β-unzipping Method in Structural System Reliability Analysis*, Computers & Structures, Vol 60, No 4, pp 619-626, 1996

Chin, R.T., Harlow, C.A., *Automated Visual Inspection: A survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 4, No 6, pp 557-73, November 1982

Chin, R.T., *Automated Visual Inspection: 1981 to 1987*, CVGIP Vol.41, pp346-381,1988

Christer, A.H., MacCallum, K.L., Kobbacy, K., Bolland, J., Hessett, C., *A Systems Model of Underwater Inspection Operations*, Journal of the Operational Research Society, Vol 40, No 6, pp 551-565, 1989

Cohn, A. G., *Deep Knowledge Representation Techniques*, Proceedings of the Fifth Technical Conference of the BCS Specialist Group on Expert Systems, Expert Systems 85, Ed. M. Merry, British Computer Workshop Series, pp 299-306, 1985

Collins, R., *The development of the ACPD and ACFM techniques at UCL*, in Studies in Applied Electromagnetics and Mechanics 8, Nondestructive Testing of Materials, Ed. by R. Collins, W.D. Dover, J.R. Bowler, K. Miya, IOS Press, 1995

Collins, R., Niemiro, A., Lewis, A.M., *Underwater Crack Measurement from Electromagnetic Field Measurement*, Advances in Underwater Technology, Ocean Science & Offshore Engineering, Volume 21: Advances in Underwater Inspection & Maintenance, Graham & Trotman, 1990

Connolly, M.P., *Reliability Estimates of the Inspection Requirements for Tubular Welded Joints*, Proceedings of the 13th Ocean, Marine and Arctic Engineering Symposium, OMAE 94, Houston Texas, US, 1994

Corkill, D., *Design Alternatives for Parallel and Distributed Blackboard systems*, Proceedings of 2nd Workshop on Blackboard Systems, pp 89-106, AAAI 88, St Paul, Minnesota, USA, 24 August 1988

Corkill, D., *Blackboard Systems*, AI Expert, September 1991

Dalane, J.I., Skjong, R., Lotsberg, I., *Optimal Fatigue Design of Offshore Structures*, Proceedings of the 9th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 90, Houston, TX, US, ASME, 1990

Davis, R., Buchanan, B., Shortliffe, E., *Production Rules as a Representation for a Knowledge-Based Consultation Program*, Artificial Intelligence, Vol 8, No 1, pp15-45, 1977(a)

Davis, R., Buchanan, B.G., *Meta-Level Knowledge: Overview and Applications*, Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-77. Cambridge, MA, USA, August 1977(b)

Davis, R., *Interactive Transfer of Expertise: Acquisition of New Inference Rules*, Artificial Intelligence, Vol 12, No 3, pp121-157, 1979

Davis, R., Lenat, D., *Knowledge Based Systems in Artificial Intelligence*, McGraw-Hill, 1982

Davis, L., (ed.), *Genetic Algorithms and Simulated Annealing*, London: Morgan Kaufmann, 1987

Dean, T.L., Wellman, M.P., *Planning and Control*, Morgan Kaufmann, 1991

DEn, *Offshore Installations: Guidance on Design and Construction*, Department of Energy, 3rd Edition, HMSO, 1984

Dharmavasan, S., *Fatigue Fracture Mechanics of Tubular Welded Y-Joints*, PhD Thesis, University of London, 1983

Dharmavasan, S., Dover, W.D., *Nondestructive Evaluation of Offshore Structures using Fracture Mechanics*, Applied Mechanics Review, Vol 41, No 2, February 1988(a)

Dharmavasan, S., Peers, S.M.C., Kam, J.C.P., *Use of AI Techniques in an Intelligent Structural Integrity System*, Europe and the Sea, International Conference on Marine Sciences and Technology in the 1990's, Hamburg, September 1988(b)

Dharmavasan, S., Reynolds, A.G., Topp, D., *The Development of a PC-based Integrated Fatigue Analysis Package for Offshore Structures*, Proceedings of the 10th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 91, Stavanger, Norway, ASME, 1991

Dharmavasan, S., Peers, S.M.C., Faber, M.H., Dijkstra, O., Cervetto, D., Manfredi, E., *Reliability Based Inspection Scheduling for Fixed Offshore Structures*, Proceedings of the 13th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 94, Houston Texas, US, ASME, 1994(a)

Dharmavasan, S., et al, *RISC Final Report, RISCREL Manual*, THERMIE Project OG/0019/90/UK-IT-NL, October 1994(b)

Dharmavasan, S., private communication, memorandum 7th November 1995

Diamantidis, D., Righetti, G., Zuccarelli, F., *Reliability Re-Qualification Criteria for Existing Jack Platforms*, Proceedings of the 10th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 91, Stavanger, Norway, Vol II, pp 213-219, ASME, 1991

Diamantidis, D., *Reliability Aspects for Design and Redesign of Offshore Structures*, Proceedings of the 5th International Offshore Mechanics and Arctic Engineering Symposium, Vol II, pp 29-36, Tokyo, 1986

Dijkstra, O.D., Straalen, IJ.J. van, *Fatigue Crack Growth Program (FAFRAM)*, TNO Building and Construction Research, Report BI-91-051, 1991

Dijkstra, O., Van Foeken, R., Dharmavasan, S., *Fracture Mechanics Limit States for Reassessment and Maintenance Planning for Offshore Structures*, Proceedings of the 13th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 94, Houston Texas, US, ASME, 1994

DnV, *Rules for the Design Construction and Inspection of Offshore Structures*, Det norske Veritas, 1982?

Dover, W.D., Dharmavasan, S., *Fatigue Fracture Mechanics Analysis Of Tubular T And Y Joints* Offshore Technology Conference, Houston, OTC 4404, 1982

Dover, W.D., et al, *Cohesive Programme of Research and Development into the Fatigue Crack Growth of Offshore Structures 1983-85, Final Report*, SERC / MTD Ltd., Department of Energy, UK, 1986

Dover, W.D., et al, *Cohesive Programme of Research and Development into the Fatigue Crack Growth of Offshore Structures 1985-87, Final Report*, SERC / MTD Ltd., Department of Energy, UK, 1988

Dover, W.D., Kare, R.F., Hall, M.S., *The Reliability of SCF Predictions using Parametric Equations: A Statistical Analysis*, Proceedings of the 10th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 91, Stavanger, Norway, ASME, 1991

Dover, W.D., Chang, E., Rudlin, J.R., *Underwater Inspection Reliability for Offshore Structures*, Proceedings of the 13th Ocean, Marine and Arctic Engineering Symposium, OMAE 94, Houston Texas, US, 1994

Dover, W.D., Rudlin, J.R., *Defect Characterisation and Classification for the ICON Inspection Reliability Trials*, Proceedings of the 15th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 96, 1996

Drawe, W.J., Reifel, M.D., *Platform Function and Types*, in *Planning and Design of Fixed Offshore Platforms*, ed. by B. McClelland, M.D. Reifel, Van Nostrand Reinhold, 1986

Dreyfuss, H.L., *From Micro-Worlds to Knowledge Representation: AI at an Impasse*, in Mind Design, ed. J. Haugeland, The MIT Press, 1981

Duchessi, P., O'Keefe, R., *A Knowledge-based Approach to Production Planning*, Journal of Operational Research, Vol 41, No 5, pp337-390,, 1990

Duda, R., Gaschnig, J., Hart, P., Model Design in the PROSPECTOR Consultant System for Mineral Exploration, Expert Systems for the Microelectronic Age, ed. D. Michie, Edinburgh University Press, 1979

Dunn, F.P., *Offshore Platform Inspection*, Proceedings of the International Symposium on The Role of Design, Inspection, and Redundancy in Marine Structural Reliability, November 14-16, 1983, National Academy Press, Washington, US, 1984

Efstathiou, J., *Anytime Heuristic Schedule Repair in Manufacturing Industry*, IEE Proceedings: Control Theory and Applications, Vol.143, No.2, pp.114-124, March 1996

Elliman, D.G., Banks, R.N., *A Net with Feedback for Recognising Deformed Patterns*, Proceedings of IEE Colloquium on "Current Issues in Neural Network Research", May 1989

Ellingworth, H., Dharmavasan, S., Peers, S., *One Year Progress Report - The AIRES EM KBS*, AIRES Report, UCL, AIRES/UCLKBS/P0090, 1992

Engelmore, R., Morgan, T., *Blackboard Systems*, Addison Wesley, 1988

Erman, L., Hayes-Roth, F., Lesser, V., Reddy, D., *The Hearsay II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty*, Computing Surveys, Vol 12, No 2, pp213-253, 1980

Evans, N., *O-O environments, not languages*, .EXE magazine, Vol 6, Issue 3, August 1991

Faber, M.H., Sorensen, J.D., Kroon, I., *Optimal Inspection Strategies for Offshore Structural Systems*, Proceedings of the 11th Ocean, Marine and Arctic Engineering Symposium, OMAE 92, Calgary, Canada, 1992

Faber, M.H., Dharmavasan, S., Dijkstra, O., *Integrated Analysis Methodology for Reassessment and Maintenance of Offshore Structures*, Proceedings of the 13th Ocean, Marine and Arctic Engineering Symposium, OMAE 94, Houston, Texas, US, 1994

Faulkner, D., Sadden, J.A., *Toward a Unified Approach to Ship Structural Safety*, The Royal Institution of Naval Architects, Paper No 3, Spring Meetings, 1978

Fikes, R., Kehler, T., *The Role of Frame-based Representation in Reasoning*, Communications of the ACM, Vol 28, No 9, September 1985

Forsell, C.,in *Prologue* (translation of excerpts from article *Ekonomi och Byggnadsvasen* by Prof C. Forsell, 1924), Lind, N.C., Structural Reliability and Codified Design, University of Waterloo, Ontario, 1970

Fox, M.S., Sadeh, N., Baykan, C., *Constrained Heuristic Search*, Proceedings of the 11th International Joint Conference on Artificial Intelligence, 1989

Fox, M.S., Smith, S.F., *ISIS - A Knowledge Base System for Factory Scheduling*, Expert Systems, Vol 1, No 1, July 1984

Frieze, P.A., *Probability based Safety Assessment of Existing and Future Offshore Structures*, Proceedings of the 8th International Conference on Offshore Mechanics and Arctic Engineering, OMAE 89, The Hague, March 19-23, 1989

Frost, R.A., *Introduction to Knowledge Based Systems*, Collins, 1987

Fuchs, H.O., Stephens, R.I., *Metal Fatigue in Engineering*, John Wiley & Sons, 1980

Fujita, M., Schall, G., Rackwitz, R., *Adaptive Reliability-based Inspection Strategies for Structures subject to Fatigue*, Proceedings of the 5th International Conference on Structural Safety and Reliability, ICOSSAR 89, 1989

Fulton, S.L., Pepe, C.O., *An Introduction to Model-based Reasoning*, AI Expert, pp 48-55, January 1990

Gale, W.A., *Statistical Applications of Artificial Intelligence and Knowledge Engineering*, The Knowledge Engineering Review, Vol 2, No 4, December 1987

Garey, M.R., Johnson, D.S., *Computers and Intractability - A Guide to the Theory of NP-Completeness, Freeman, NY,* 1979

Gaschnig, J., *A Problem Similarity Approach to Devising Heuristics: First results*, Proceedings of 6th International Joint Conference on Artificial Intelligence, North-Holland Pub. Co., 1981

Georgeff, M.P., *Planning,* Annual Review of Computer Science, 1987

Gierlinski, J.T., Sears, R., Shetty, N.K., *Integrity Assessment of Fixed Offshore Platforms - A Case Study using RASOS*, Proceedings of the 12th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 93, Glasgow, UK, ASME, 1993

Gnedenko, B.V., Ushakov, I.A., *Probabilistic Reliability Engineering* , Wiley Interscience, 1995

Gordon, J.E., *Structures (or why things don't fall down)*, Penguin Books, 1978

Gordon, R., *Optimum Component Redundancy for Maximum System Reliability*, Operations Research, Vol 5, 229-243, 1957

Goyet, J., Faber, M.H., Paygnard, J.-C., Maroini, A., *Optimal Inspection and Repair Planning: Case Studies Using IMREL Software*, Proceedings of the 13th Ocean, Marine and Arctic Engineering Symposium, OMAE 94, Houston, Texas, US, 1994

Gresnigt, A.M., Van Foeken, R., *Strength and Deformation Capacity of Pipelines Loaded by Local Loads and Bending*, Proceedings of the Zeepipe Conference, Oostende, Netherlands, 1990

Gresnigt, A.M., Dijkstra, O.D., Van Rongen, H.J.M., *Design of Pipelines in High Strength Steel*, Proceedings of the 4th International Offshore and Polar Engineering Conference, ISOPE-94, Osaka, Japan, 1994

Griffiths, D.G., Mamdani, E.H., Efstathiou, H.J., *Fuzzy Sets and Expert Systems*, Anal of Fuzzy Inference, CRC Press Inc, Boca Raton, FL, USA, 1987

Guilfoyle, C., *Ten Minutes to Lay*, Expert System User, August 1986

Hardman, D.K., Ayton, P., *Arguments for Qualitative Risk Assessment: the StAR Risk Adviser*, Expert Systems, Vol 14, No 1, February 1997

HSE, *The Tolerability of Risk from Nuclear Power Stations*, Health & Safety Executive, HMSO, 1992

Holdbrook, S.J., Dover, W.D., *The Stress Intensity Factor for a Deep Surface Crack in a Finite Plate*, Engineering Fracture Mechanics, Vol 12, No 3, pp 347-64, 1979

IMP communication at meeting, Instituto Mexicano del Petroleo, Mexico DF, Mexico, 1992 (a)

IMP communication, Report, Instituto Mexicano del Petroleo, Mexico, 1992 (b)

IMP communication on procedures carried out at IMP, Instituto Mexicano del Petroleo, Mexico, 1996

IntelliCorp, *ProKappa Development Software*, Beta Version for AIX, IntelliCorp Inc., US, 1991

IntelliCorp, *Kappa PC*, Version 2.0, IntelliCorp Inc., US, 1992

Jardine, A.K.S., *Maintenance, Replacement and Reliability*, Pitman. 1973

Jennings, N.R., Mamdani, E.H., Corera, J.M., Laresgoiti, I., Perriollat, F., Skarek, P., Varga, L.Z., *Using Archon to Develop Real-World DAI Applications, Part 1*, IEEE Expert, December 1996

Jiang, K., Seneviratne, L.D., Earles, S.W.E., *Assembly Scheduling for an Integrated Two-Robot Workcell*, Robotics and Computer-Integrated Manufacturing, Vol.13, No.2, pp.131-143, June 1997

Johnson, L. Keravnou, E.T, *Expert Systems Architectures*, Kogan Page, 1985

Jones, S., *Graphical interfaces for knowledge engineering: an overview of relevant literature*, The Knowledge Engineering Review, Vol 3, No 3, pp 221-245, September 1988

Kam, J.C.P., Dover, W.D., *Structural Integrity of Welded Tubular Joints in Random Load Fatigue combined with Size Effect*, Proceedings of the Integrity of Offshore Structures International Conference, IOS 87, Glasgow, September 1987

Kam, J.C.P., Dover, W.D., *Fast Fatigue Assessment for Offshore Structures under Random Stress History*, Proceedings of the Institution of Civil Engineers, Part 2, December 1988

Kam, J.C.P., *The Reliability Assessment Of Offshore Structures Under The Influence Of Fatigue Crack Growth*, Proceedings of the 10th International Symposium on the Advances in Reliability Technology, Bradford, April, 1988

Kam, J.C.P., *The Efficient Maintenance of Offshore Structural Integrity using Reliability Analysis*, Quality and Reliability Engineering International, Vol 5, pp 221-228, 1989(b)

Kam, J.C.P., *Structural Integrity of Offshore Tubular Joints Subject to Fatigue*, PhD Thesis, University of London, 1989(a)

Kam, J.C.P., *Recent Development in the Fast Corrosion Fatigue Analysis of Offshore Structures Subjected to Random Wave Loading*, International Journal of Fatigue, 1990

Katsoulakos, P.S., Hornsby, C.P.W., *Expert Systems and Marine Applications*, The Institute of Marine Engineers, October 18, 1988

Kemp, P., Saran, C., *Upper, Middle or Working Classes?*, .EXE magazine, Vol 6, Issue 3, August 1991

Khong, V.H., Lucia, A.C., *RAMINO - A Knowledge Based System and a Co-ordinator of Numerical Models for Structural Reliability Assessment*, Proceedings of the 10th International Workshop on Expert Systems and their Applications, Avignon, 1990

Kirk, I., Lewcock, A., *Neural Networks - an Introduction*, Insight, Vol 37, No 1, 1995

Kirkemo, F., *Applications of Probabilistic Fracture Mechanics to Offshore Structures*, Applied Mechanics Review, Vol 41, No 2, February 1988

Kobbacy, K.A.H., *Use of knowledge-based systems in evaluation and enhancement of maintenance routines*, International Journal of Production Economics, Vol.24, No.3, pp.243-248, Mar 1992

Kobbacy, K.A.H.; Proudlove, N.C.; Harper, M.A., *Towards an Intelligent Maintenance Optimization System*, Journal of the Operational Research Society, Vol 46, No 7, pp831-853, July 1995

Koenig, A., Crochon, E., *Temporal Reasoning in TRAM*, Proceedings of The Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE-89, Tullahoma, USA, ACM, New York, NY, USA, 1989

Koenig, A., Crochon, E., Antoniakis, M., *TRAM Manual*, LETI, 1990

Korf, R.E., *Planning As Search: A Quantitative Approach*, Artificial Intelligence, Vol 33, pp65-88, 1987

Kosko, B., *Neural Networks and Fuzzy Systems - A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, 1992

Kumar, V., *Algorithms for Constraint Satisfaction Problems: A Survey*, AI Magazine, Vol 13, No 1, pp32-44, 1992

Lang, P., *Economic Future of North Sea Gas Fields*, Journal of the Operational Research, Vol 41, No 2, pp119-123, 1990

Langdon, A., Ahmad, K., Frieze, P.A., *An Expert System for Offshore Structure Inspection and Maintenance*, Artificial Intelligence Techniques and Applications for Civil and Structural Engineers, Civil-Comp Press, Edinburgh, UK, 1989

Laughton, M.A., Hawken, A., Chui, D., Ekwue, A.C., Macqueen, J.F. Taylor, A., *Fuzzy Techniques for Voltage Control on the NGC System*, Proceedings of the International Conference on CONTROL '94, Coventry, UK, March 1994

Laurent, J.P., *Types of Control Structure in Expert Systems*, The Knowledge Engineering Review, Vol 2, No 2, June 1987

Le Pape, C., *Classification of Scheduling Problems and Selection of Constraint-Based Techniques*, Colloquium on Advanced Software Technologies for Scheduling, Institute of Electrical Engineers, London, UK, 26 April, 1993

Long, D., *A Review of Temporal Logic*, The Knowledge Engineering Review, Vol 4, No 2, pp 141-162, June 1989

Lotsberg, I., Marley, M.J., *In-service Inspection Planning for Steel Offshore Structures using Reliability Methods*, Proceedings of Behaviour of Offshore Structures Conference, BOSS 92, London, 7-10 July 1992

Ma, C.N., Kam, J.C.P., *Crack Shape Evolution in Tubular Welded Joints*, NDT&E International, Vol 24, No 6, December, 1991

Maddox, S.J., *Fatigue Strength of Welded Structures*, 2nd ed, Abington Publishing, 1991

Madsen, H.O., Skjong, R., Kirkemo, F., *Probabilistic Fatigue Analysis of Offshore Structures - Reliability Updating through Inspection Results*, Proceedings of the Integrity of Offshore Structures International Conference, IOS 87, Glasgow, September 1987(a)

Madsen, H.O.,Krenk, S., Lind, N.C., *Methods of Structural Safety*, Prentice-Hall, 1987(b)

Maes, P., *Computational Reflection*, The Knowledge Engineering Review, Vol 3 No 1, pp 1-19, 1988

Maher, M.L., Gomez de Silva Garza, A., *Developing Case-based Reasoning for Structural Design*, IEEE Expert, June 1996

Mallat, S.G., *A Theory for Multiresolution Signal Decomposition: the Wavelet Representation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 11, 674-691, 1989

Mamdani, A., Efstathiou, J., Pang, D., *Inference Under Uncertainty*, Proceedings of the 5th Technical Conference of the British Computer Society Specialist Group for Expert Systems, Expert Systems 85, Cambridge University Press, 1985

McCarthy, J., *Programs with Common Sense*, Semantic Information Processing, ed. M. Minsky, The MIT Press, 1968

McClelland, B., Reifel, M.D., *Planning and Design of Fixed Offshore Platforms*, Van Nostrand Rheinhold, 1986

McDermott, J., *R1: The Formative Years*, AI Magazine, Vol 2, No 2, pp21-29, 1981

McMahon, C.A., Banerjee, S., Sims Williams, J.H., Devlukia, J., *Hypertext and Expert Systems Application in Fatigue Assessment and Advice*, Automation in Fatigue and Fracture: Testing and Analysis, ASTM STP1231, ed. C. Amzallug, American Society for the Testing of Materials, Philadelphia, 1994

McNab, A., Dunlop, I., *A Review of Artificial Intelligence Applied to Ultrasonic Defect Evaluation*, Insight Vol 37, No 1, January 1995

Mettrey, W., *A Comparative Evaluation of Expert System Tools*, Computer, February 1991

Michael, D.H., Collins, R., *Recent Studies in Modelling for the A C Field Measurement Technique*, Review of Progress in Quantitative Nondestructive Evaluation, Vol. 6A, Plenum Press, 1987

Michael, D.H., Lewis, A.M., McIver, M., Collins, R., *Thin-skin Electromagnetic Fields in the Neighbourhood of Surface-breaking Cracks in Metals*, Proceedings of the Royal Society London, A (1991) 434, 587-603, 1991

Minsky, M.A. , *A Framework for Representing Knowledge*, in The Psychology of Computer Vision, ed. P.H. Winston, McGraw-Hill, New York, 1972

Miyashita, K., *Case-based Knowledge Acquisition for Schedule Optimisation*, Artificial Intelligence in Engineering, Vol 9, pp27-287, 1995

Moan, T., Vardel, O.T, Hellevig, N.-C., Skjoldli, K., *In-Service Observations of Cracks in North Sea Jackets, A Study on Initial Crack Depth and POD Values*, Proceedings of 16th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 97, ASME 1997

Moyzis Jr., J.A., Forney Jr., D.M., *Increased Reliability - A Critical NDE Research Goal*, Review Progress in Quantitative Nondestructive Evaluation, Vol 1, Plenum Press, pp 7-11, 1982

MTD, *Underwater Inspection of Steel Offshore Installations: Implementation of a New Approach*, The Marine Technology Directorate Limited, MID Ltd Publication 89/104, 1989

Navinchandra, D., Sriram, D., Logcher, R.D., *GHOST: A Project Network Generator*, Journal of Computing in Civil Engineering, Vol 2, No 3, July 1988

Newell, A., Simon, H.A., *GPS, A Program that Simulates Human Thought*, in Computers and Thought, ed. E.A. Feigenbaum & J. Feldman, pp279-293, Oldenbourg, 1963

Newell, A., Steier, D., *Intelligent Control of External Software Systems*, Artificial Intelligence in Engineering Vol 8, pp3-21,1993

Newman, J., Raju, I., *An Empirical Stress Intensity Factor for the Surface Crack*, Engineering Fracture Mechanics, Vol 22, No 5, pp185-192, 1981

Nicholson, R.W., Fraser, J.R., Stannard, D.M., *An Analytically Based Approach to the Optimisation of Underwater Inspection of Steel Platforms*, Proceedings of the IRM-AODC 86, 3-6 November, Aberdeen, UK, 1986

Nii, H.P., Aiello, N., *AGE (Attempt to Generalize): A Knowledge-Based Program for Building Knowledge-Based Programs*, Proceedings of the 6th International Joint Conference on Artificial Intelligence, IJCAI-79, pp645-655, Cambridge, MA, USA, 1979

Nii, H.P., *Part 1: Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures*, The AI Magazine, pp 38-53, Summer 1986 (a)

Nii, H.P., *Part 2: Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures*, The AI Magazine, pp 82-106, August 1986 (b)

Nolan, P.J., McCarthy, M.A., *AI Frame-Based Simulation in System Dynamics*, Applications of AI in Engineering, Proceedings of 1st International Conference, Southampton University, UK, A Computational Mechanics Publication, Springer-Verlag, April 1986

Paris, P.C., Erdogan, F., *A Critical Analysis of Crack Propagation Laws*, Transactions of ASME, Journal of Basic Engineering, Vol 85, No 4, p 528, 1963

Parrod, Y., Valera, S., *Optimum-AIV, a Planning Tool for Spacecraft AIV*, Preparing for the Future, Vol 3, No 3, 1993

Pau, L., *Knowledge Representation Approaches in Sensor Fusion*, Automatica, Vol 25, No 2, pp 207-214, 1989

Pearl, J., *Heuristics - Intelligent Search Strategies for Computer Problem-Solving, Addison-Wesley,* 1984

Peers, S.M.C., Dharmavasan, S., Kam, J.C.P., Dover, W.D., *A Rational Inspection Scheduling Philosophy for Fixed Offshore Structures*, Proceedings of the 11th Ocean, Marine and Arctic Engineering Symposium, OMAE 92, Calgary, Canada, 1992

Peers, S.M.C., *A Frame-Based Repository of Computer Programs*. Application of Artificial Intelligence in Structural Engineering IV, Proceedings of the 4th Workshop of the EG-SEA-AI, Finland, September, 1997

Pook, L.P., *The Role of Crack Growth in Metal Fatigue*, The Metals Society, London, 1983

POSC, *Software Integration Platform Specification, Base Computer Standards, Version 1.0*, Petrotechnical Open Software Corporation, 1991

Price, C., Hunt, J., *Using Qualitative Reasoning for Diagnostic Applications*, IEE Colloquium on Expert Systems for Fault Diagnosis in Engineering Applications, London, 20 April 1989

Prosser, P., *The Future of Scheduling - DAI?*, Colloquium on Advanced Software Technologies for Scheduling, Digest No 1993/163, Institute of Electrical Engineers, London, 26 April 1993

Puget, J.-F., *Object Oriented Constraint Programming for Transportation Problems*, Colloquium on Advanced Software Technologies for Scheduling, Institute of Electrical Engineers. London, UK, 26 April 1993

Raiffa, H., Schlaifer, R., *Applied Statistical Decision Theory*, Harvard University, Boston, Massachusetts, US, 1960

Raima, *Raima Data Manager, User and Technical Manuals*, Raima Corporation, Bellevue, US, 1991

Rajagoplan, H.S., Grandhi, R.V., *Reliability based Structural Analysis and Optimization in X Window Environment*, Computers & Structures, Vol 60, No 1, pp1-10, 1996

Rajasekaran, S., Febin, M.F., Ramasamy, J.V., *Artificial Fuzzy Neural Networks in Civil Engineering*, Computer & Structures, Vol 61, pp261-302, 1996

Ray, T., Gokarn, B.P., Sha, O.P., *Neural Networks Applications in Naval Architecture and Marine Engineering*, Artificial Intelligence in Engineering, Vol 1, pp 213-226,1996

RBI, *Risk-based Inspection - Development of Guidelines, Vol 1 General Document*, prepared by Research Task Force on Risk-Based Inspection Guidelines, CRTD-Vol.20-1, American Society of Mechanical Engineers, 1989

Registro Italiano Navale, *RISC Section 2 Theory Chapter 1 Analysis Procedure for IRM*, RISC Document, NDE Centre, UCL, 1994

Reichgelt, H., *Logics for Reasoning about Knowledge and Belief*, The Knowledge Engineering Review, Vol 4, No 2, pp 119-139, June 1989

Reiter, R., *On Reasoning By Default*, Proceedings of Theoretical Issues in Natural Language Processing, TINLAP-2, pp210-218, Illinois, 1978

Rivers, C., *Optimising Inspection and Maintenance - Theory and Practice*, Proceedings of the IRM-AODC 86, 3-6 November, Aberdeen, UK, 1986

Rojiani, K.B., Bailey, G.L., *Reliability Based Optimum Design of Steel Buildings*, in New Directions in Optimum Structural Design, ed. Atrek et al, Wiley & Sons, 1984

Rudlin, J.R., *Investigation of Underwater MPI Procedures in the ICON Project*, Insight, Vol 38, No 6, June 1996

Rudlin, J.R., Dover, W.D., *The ICON Project - Data for Underwater Inspection*, Insight, Vol 38, No 6, June 1996

Rudlin, J.R., Wolstenholme, L.C., *Development of Statistical Probability of Detection Models using Actual Trial Inspection Data*, British Journal of NDT, Vol 34 No 12, December 1992

Rummel, W.D., Christner, B.K., Long, D.L., *Methodology for Analysis and Characterization of Nondestructive Inspection Capability Data*, Review of Progress in Quantitative Nondestructive Evaluation, Vol 7B , 1988

Rychener, M.D., *Expert Systems for Engineering Design*, Expert Systems, Vol 2, No 1, pp 31-44, January 1985

Saffiotti, A., *An AI View of the Treatment of Uncertainty*, The Knowledge Engineering Review, Vol 2, No 2, June 1987

Saldanha Peres, J.E.C., Rogerson, J.H., *The Use of Probabilistic Fracture Mechanics in Devising Quality Control Policies in the Fabrication Industry*, Reliability Engineering, Vol 8, pp149-164, 1984

Schank, R.C., Abelson, R.P., *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*, Lawrence Erlbaum, 1977

Sharp, J., Kam, J.C., Birkinshaw, M., *Review of Criteria for Inspection and Maintenance of North Sea Structures*, Proceedings of the 12th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 93, Glasgow, UK, ASME, 1993

Shetty, N.K., *Selective Enumeration Method for Identification of Dominant Failure Paths of Large Structures*, Proceedings of the 13th Ocean, Marine and Arctic Engineering Symposium, OMAE 94, Houston Texas, US, ASME, 1994

Shortliffe, E., *Consultation Systems for Physicians*, Proceedings of Canadian Society for Computational Studies of Intelligence, University of Victoria, Victoria BC, 1980

Shyamsunder, M.T., Rajagopalan. C., Ray, K.K. Baldev Raj, *A Comparative Study of Conventional and Artificial Neural Network Classifiers for Eddy Current Signal Classification*, Insight, Vol 37, No 1, 1995

Smith, S.F., Ow, P.S., Potvin, J.-Y., Muscettola, N., Matthys, D.C., *An Integrated Framework for Generating and Revising Factory Schedules*, Journal of the Operational Research Society, Vol 41, No 6, 1990

Snell, R.O., *Outline of the ISO Offshore Structures Code and its Contribution to Reliability Assessment,* Proceedings of SUT International Conference, API-RP 2A-LRFD - Its Present and Future Role in Offshore Safety Cases, 24 November 1993

Sorensen, J.D., Faber, M.H., Rackwitz, R., Thoft-Christensen, P., *Modelling in Optimal Inspection and Repair*, Proceedings of the 10th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 91, Stavanger, Norway, Vol II, ASME, 1991

Spanner Sr., J.C., *Human Factors Impact on NDE Reliability*, Review of Progress in Quantitative Nondestructive Evaluation, Vol 7B , 1988

Sriram, D., Maher, M.L., *The Representation and Use of Constraints in Structural Design,* Proceedings of 1st International Conference Applications of AI in Engineering Problems, Vol 1, Southampton University, Springer-Verlag, April 1986

Stanojevic, M., Vranes, S., Velasevic, D., *Using Truth Maintenance Systems - A Tutorial,* IEEE Expert, pp46-56, December 1994

Stefik, M., *Planning with Constraints (MOLGEN Part 1)*, Artificial Intelligence, Vol 16, No2, 1981

Stefik, M., Bobrow, D., *Object-Oriented Programming: Themes and Variations*, The AI Magazine, 1984

Stewart, G., Efthymiou, M., Vugts J.H., *Ultimate Strength and Integrity Assessment of Fixed Offshore Structures*, Proceedings of the International Symposium on Behaviour of Offshore Structures, BOSS 88, 1988

Stiansen, S.G., *Interrelation between Design, Inspection and Redundancy in Marine Structures,* Proceedings of International Symposium The Role of Design, Inspection and Redundancy in Marine Structural Reliability, National Academic Press, November 1983

Sticklen, J., Smith Jr., J.W., Chandrasekaran, B., Josephson, J.R., *Modularity of Domain Knowledge*, International Journal of Expert Systems, Vol 1, No 1, pp1-15, 1987

Stone, J.R., Blockley, D.I., Pilsworth, B.W., *Towards Machine Learning from Case Histories*, Civil Engineering Systems, Vol 6, No 3, pp129-135, 1989

Strange, P.G., *Basic Brain Mechanisms: A Biologist's View of Neural Networks,* Proceedings of IEE Colloquium on "Current Issues in Neural Network Research", May 1989

Suganan, N, *RISC Final Report Section 2 Chapter 5.2 SCF Database,* NDE Centre, UCL, 1994

Tao, S., Song, Z.H., Selviah, D.R., Midwinter, J.E., *Spatioangular-Multiplexing Scheme for Dense Holographic Storage*, Applied Optics, Vol.34, No.29, pp.6729-6737. Oct 10 1995

Tate, A., *Generating Project Networks*, Proceedings of the International Joint Conferences on Artificial Intelligence, IJCAI-77, pp888-893, Cambridge, MA, USA, August 1977

Tate, A.,Hendler, J., Drummond, M., *A Review of Planning Techniques*, Knowledge Engineering Review, Vol 1, No 2, pp4-17, June 1985

Tello, E.R., *Object-oriented Programming for Artificial Intelligence*. Addison-Wesley, 1989

The Offshore Installations (Construction and Survey) Regulations, 1974, Statutory Instrument Number 289, HMSO

Thoft-Christensen, P., Baker, M.J., *Structural Reliability Theory and its Applications*, Springer-Verlag, 1982

Thoft-Christensen, P., Murotsu, Y., *Application of Structural Systems Reliability Theory*, Springer-Verlag, 1984

Thornton, C., Du Boulay, B., *Artificial Intelligence Through Search*, Intellect Books, 1992

Thorpe, T.W., *A Simple Model of Fatigue Crack Growth*, Department of Energy, Offshore Technology Report, OTH 86 225, HMSO, 1986

Topp, D., Dover, W.D., *Offshore Inspection/Repair - A Changing Philosophy*, Asian Conference on Inspection, Repair & Maintenance for Offshore & Marine Industries, Singapore, February 1985

Topping, B.H.V., Kumar, B., *Knowledge Representation and Processing for Structural Engineering Design Codes*, Engineering Applications of Artificial Intelligence, Vol 2 (3), Swansea, pp214-227, September 1989

TSC, *FACTS Technical Reference*, Technical Software Consultants. Milton Keynes, UK, 1990

Turkstra, C. J., *Applications of Bayesian Decision Theory*, in Structural Reliability and Codified Design, Ed. N.C. Lind, 1969

Vardel, O.T, Moan, T, *Predicted versus Observed Fatigue Crack Growth, Validation of Probabilistic Fracture Mechanics Analysis of Fatigue in North Sea Jackets*, Proceedings of 16th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 97, ASME 1997

Wall, M., Wedgwood, F.A., *NDT - Value for Money*, Insight, Vol 36, No 10, October 1994

Weiss, V., *Towards Failure Analysis Expert Systems*, ASTM Standardisation News, Vol 14, No 4, pp 30-34, April 1986

Willoughby, A.A., Laures, J.-P., *Development of a Knowledge-Based System for Assessing Flaws in Welded Structures*, Proceedings of Expert Systems 90 Conference, September 1990

Windsor, C.G., *Can we train a Computer to be a skilled Inspector?*. Insight, Vol 37, No 1, January 1995

Winkworth, W.J., Fisher, P.J., Griffiths, M.S., *The Application of API RP 2A-LRFD to North Sea Platforms - An Overall Review*, Proceedings of SUT International Conference, API-RP 2A-LRFD - Its Present and Future Role in Offshore Safety Cases, 24 November 1993

Winston, P.H., *Artificial Intelligence*, Addison-Wesley, 1977

Wirsching, P.H., *Probability Based Fatigue Design Criteria For Offshore Structures, Final Report*, API PRAC 15, American Petroleum Institute, January, 1985

Wirsching, P.H., *Fatigue Reliability for Offshore Structures*, Journal of Structural Engineering, ASCE, Vol 110, No 10, Oct 1984

Wolfram, J., *The Effects Of Fatigue Cracking Upon The Reliability Of Tubular Members Of Offshore Steel Structures*, Proceedings of 5th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 85, Tokyo, 1986

Wu, Y., Du, R., *Feature Extraction and Assessment using Wavelet Packets for Monitoring of Machining Process*, Mechanical Systems and Signal Processing, Vol 10, No 1, 29-53, 1996

Ximenes, M.C., Mansour, A., *Fatigue System Reliability Including Inspection Updating*, Proceedings of the 10th International Offshore Mechanics and Arctic Engineering Symposium, OMAE 91, Stavanger, Norway, Vol II, pp203-211, ASME, 1991

Yagin, T., *A Predicate Logical Method for Modelling Design Objects*, AI in Engineering, Vol 4, No 1, Jan 1989, pp 41-53

Yang, Q., *Intelligent Planning - A Decomposition and Abstraction Based Approach*, Springer 1997

Zadeh, L., *On the Treatment of Uncertainty in AI*, The Knowledge Engineering Review, Vol 3 No 5 1, pp1-19, 1988

Zadeh, L., *Calculus of Fuzzy Restrictions*, US-Japan Seminar on Fuzzy Sets and their Applications to Cognitive and Decision Processes, Berkeley, CA, US, 1974

Zaniolo, C., *Object-Oriented Programming in PROLOG*, Proceedings of International Logic Symposium, IEEE, 1984

Zaniolo, C., *The Representation and Deductive Retrieval of Complex Objects*, Proceedings of VLDB 85, Stockholm, 1985

Zhou, J., Michael D.H., Collins, R., *Half-Space Induction by a Rectangular Coil with Rounded Corners*, Review of Progress in Quantitative Nondestructive Evaluation, Vol. 13, Plenum 1993

Zhou, J., Collins, R., *AIRES EM Theory Final Report*, AIRES Report, UCL, AIRES/UCLEM/P0096, 1994

Zimmerman, J.J., Banon, H., *System Fatigue Reliability and Inspection Planning for Offshore Platforms,* Proceedings of the 13th Ocean, Marine and Arctic Engineering Symposium, OMAE 94, Houston Texas, US, ASME, 1994

APPENDIX

PUBLISHED PAPERS

Dharmavasan, S., Peers, S.M.C., Kam, J.C.P., *Use of AI Techniques in an Intelligent Structural Integrity System*, Europe and the Sea, International Conference on Marine Sciences and Technology in the 1990's, Hamburg, September 1988

Peers Chapman, S.M.C., Kam, J.C.P., *The Development of Reliability Fracture Mechanics Method for the Rational Fatigue Assessment of the Safety of Offshore Structures*, Proceedings of International Symposium, COMPUMAT-89, Paris, France, May 1989

Peers Chapman, S.M.C., Dharmavasan, S., *Knowledge Based Assistant for Design of Offshore Structures*, Proceedings of International Symposium on Expert Systems, EXPERSYS-89, Paris, France, October 1989

Peers, S.M.C., Dharmavasan, S., Kam, J.C.P., Dover, W.D., *A Rational Inspection Scheduling Philosophy for Fixed Offshore Platforms*, Proceedings of 11th International Symposium Offshore Mechanics and Arctic Engineering, ASME, Calgary, Canada, June 1992

Dharmavasan, S., Peers, S.M.C., *General Purpose Software for Fatigue Testing*, Automation in Fatigue and Fracture: Testing and Analysis, ASTM STP1231, ed. C. Amzallug, American Society for the Testing of Materials, Philadelphia, 1994

Peers, S.M.C., Tang, M.X., Dharmavasan, S., *Knowledge Based Approach to Inspection Planning for Offshore Structures*, Proceedings of 13th International Symposium Offshore Mechanics and Arctic Engineering, OMAE-94, ASME, Houston, USA, March 1994

Dharmavasan, S., Peers, S.M.C., Faber, M.H., Dijkstra, O., Cervetto, D., Manfredi, E., *Reliability Based Inspection Scheduling for Fixed Offshore Platforms*, Proceedings of 13th International Symposium Offshore Mechanics and Arctic Engineering, OMAE-94, ASME, Houston. USA, March 1994

Tang, M.X., Dharmavasan, S., Peers, S.M.C., *Use of Knowledge Based Systems for Rational Reliability Analysis Based Inspection and Maintenance Planning for Offshore Structures*, Proceedings of 4th International Symposium of Offshore and Polar Engineering, ISOPE-94, ASME, Osaka, Japan, April 1994

Tang, M.X., Peers, S.M.C., Dharmavasan, S., *Development of a Knowledge Based System for Offshore Platform Inspection Scheduling*, Proceedings of 7th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 94, Austin Texas, USA, May 1994

Dharmavasan, S., Peers, S.M.C., Tang, M.X., *Integration of Reliability Analysis within a Knowledge Based Framework*, Proceedings of International Conference on Behaviour of Offshore Structures, BOSS 94, Boston, USA, July 1994

Peers, S.M.C., Tang, M.X., , Dharmavasan, S., *A Knowledge Based Scheduling System for Offshore Structure Inspection*, Applications of Artificial Intelligence in Engineering IX, Proceedings of 9th International Conference, AIEng '94, Pennsylvania, USA, July 1994

Peers, S M C, Dharmavasan, S, Ellingworth, H, *Automated Image Recognition using Expert Systems (AIRES) - ElectroMagnetic Knowledge Based System (EM-KBS), Final Report*, NDE Centre, July 1994

Dharmavasan, S., Peers, S.M.C., Faber, M.H., Dijkstra, O., Cervetto, D., Manfredi, E., *RISC System, Demonstration software,* delivered to RISC project partners and sponsors, December 1994

Dharmavasan, S., Peers, S.M.C., Faber, M.H., Dijkstra, O., Cervetto, D., Manfredi, E., *Reliability Based Inspection Scheduling for fixed Offshore Structures (RISC) Final Report*, NDE Centre, December 1994

Peers, S, Dharmavasan, S., Collins, R., *Advanced Software for Automatic Defect Classification and Characterisation*, Studies in Applied Electromagnetics and Mechanics 8, Nondestructive Testing of Materials, Ed. By R.Collins, W.D. Dover, J.R. Bowler and K.Miya, IOS Press, 1995

Dharmavasan, S., Peers, S.M.C., *Inspection, Maintenance and Repair Planning and Scheduling for Tubular Joints using Reliability Methods*, Fatigue of Offshore Structures, Ed. A G Madhava Rao and W D Dover, Oxford University Press, 1996

Peers, S.M.C., *A Frame-Based Repository of Computer Programs*, Application of Artificial Intelligence in Structural Engineering IV, Proceedings of the 4th Workshop of the EG-SEA-AI, Finland, September, 1997

Peers, S.M.C., *Scheduling of Inspection, Repair and Maintenance for Fixed Offshore Structures,* Application of Artificial Intelligence in Structural Engineering IV, Proceedings of the 4th Workshop of the EG-SEA-AI, Finland, September, 1997

Peers, S.M.C., *A KBS for Automatic Defect Classification and Characterisation*, Application of Artificial Intelligence in Structural Engineering IV, Proceedings of the 4th Workshop of the EG-SEA-AI, Finland, September, 1997

Peers, S.M.C., *Knowledge Representation in a Blackboard System Approach for Sensor Data Interpretation*, Lecture Notes in Artificial Intelligence 1415, Methodology and Tools in Knowledge-Based Systems, ed. J. Mira, A. Pasqual del Pobil, M. Ali, Springer, 1998

Peers, S.M.C., *A Blackboard System Approach to Electromagnetic Sensor Data Interpretation*, Expert Systems, Vol 15, No 3, pp197-215, August 1998

Peers, S.M.C., *Searching for Rational Inspection Schedules for Fixed Offshore Platforms*, submitted to Engineering Applications of Artificial Intelligence, 1998